Algorithmic Tools for Data-Oriented Law Enforcement

Tim Cocx

The work in this thesis has been carried out under the auspices of the research school IPA (Institute for Programming research and Algorithmics).

This research is part of the DALE (Data Assistance for Law Enforcement) project as financed in the ToKeN program from the Netherlands Organization for Scientific Research (NWO) under grant number 634.000.430.

Cover design: Daphne Swiebel en Tim Cocx.

ISBN: 978-90-9024805-9

Algorithmic Tools
for
Data-Oriented Law Enforcement

Proefschrift

ter verkrijging van
de graad van Doctor aan de Universiteit Leiden,
op gezag van de Rector Magnificus prof. mr. P.F. van der Heijden,
volgens besluit van het College voor Promoties
te verdedigen op woensdag 2 december 2009
klokke 11.15 uur

door

Tim Kristian Cocx
geboren te Amstelveen
in 1979

Promotiecommissie

Promotor:	prof. dr. J.N. Kok	
Co-promotor:	dr. W.A. Kosters	
Overige leden:	prof. dr. Th. Bäck	
	prof. dr. H. Blockeel	(Katholieke Universiteit Leuven)
	prof. dr. S. Haring	
	dr. M.-F. Moens	(Katholieke Universiteit Leuven)
	prof. dr. A.P.J.M. Siebes	(Universiteit Utrecht)

Contents

Chapter 1

Introduction

In the wake of the data explosion of the late 1990s a research area has evolved from statistics and computer science. The main goal of this form of computer guided data analysis, known as *Data Mining* [19] or *Knowledge Discovery in Databases* (KDD), is to extract knowledge from an, often large, collection of data, combining elements of statistics [8], database technology, machine learning [2], artificial intelligence [15, 9, 10] and visualization, within its varying approaches.

The increase in capabilities of information technology of the last decade has led to a large increase in the creation of data [11], as a by-product of corporate and governmental administration or resulting from scientific analyses. Although most of this data has a purpose of its own, for example customer management, tax refund declaration archives and DNA analysis, data mining software aims to aggregate a higher level of information, knowledge, from this data, through the automatic discovery of (underlying) patterns, behavioral classes or (communication) networks. Respectively, convenient knowledge can be gained about customer behavior, tax evasion patterns or discriminating DNA strings that define human biological disorders. Naturally, the algorithms compiled for such purposes are often easily transferable to other domains with the purpose of performing similar tasks.

One of these potential application domains is that of law enforcement. As a an authority that is strictly governed by judicial constraints, the adaptation from paper based archives to a digitally oriented data-infrastructure has been relatively slow, but in recent years, especially since the tragic events of 9/11, law enforcement agencies have begun to bridge this gap by investing more resources in suitable and uniform information systems [12]. Just like in other areas, this change has led to an abundance of data, that is probably suitable for data mining purposes. This thesis describes a number of efforts in this direction and reports on the results reached on the application of its resulting algorithms on actual police data. The usage of specifically tailored data mining algorithms is shown to have a great potential in this area, which forebodes a future where algorithmic assistance in “combating” crime will be a valuable asset.

1.1 Data Mining

Within the area of computer guided statistical data analysis a clear distinction is made between data and knowledge, but the relation between these two types of information is subtle. Although it is clear that data in itself has great value within almost all aspects of modern society, most data collected remains unused as it is never called upon by its creator, its usability depending on independent and very specific cases. For example customer records are only relevant if this particular customer has a complaint about one of the transactions and since the majority of transactions happen without any problems, these records can lie dormant forever. Also, data in itself is sometimes irrelevant if no lessons can be learned from it. For example, obtaining the entire human genome can be quite useless if researchers were unable to use this data for the identification or prevention of disorders. Knowledge, being a result of learning and reasoning on certain perception, has an inherit value and is often usable for progression. Taking this into account the relation can be described as data being the building blocks from which knowledge can be derived.

Obtaining knowledge from perception has been an activity performed by humans alone, for a long time, but the ever growing amount of data, beyond the limits of human perception, and the improvements in information technology have opened the door for computer involvement in this process as both a possibility and a necessity. This process of computer guided data analysis consists of several sub-steps:

1. Data collection
2. Data preparation
3. Data analysis
4. Result visualization

There is some general confusion about the usage of the term data mining to describe either this entire process [19] or the third sub-step alone [6], however, this distinction is only of interest in theoretical discourse, rather than being relevant in practice: If wrong or incomplete data is collected, it is prepared poorly or the results are not or counter intuitively visualized to the user, the data analysis efforts are hardly relevant. Therefore, data mining is usually seen as the process that “extracts implicit, previously unknown and potentially useful information from data” [19].

Naturally, data exists in different forms, but in the scope of data mining, one is limited to data stored in digital form. However, a distinction can still be made, within this subcategory, between structured data and data that is stored without any constraints in an unstructured way. Usually and preferably, corporate data is structured into databases that allow for easy manipulation and searching through the association of an object with attribute-value pairs. Data stored in such a way is highly structured and allows for easy data collection and preparation. A drawback to this approach is that data that does not adhere to this predefined structure is stored with difficulties, including empty attributes or mismatches in attribute types. Next to relational database oriented storage, a lot of information is stored in an unstructured way, like, for example, websites, policy documents, meeting transcripts, etc. Data mining methods that deal with this kind of information first attempt to structure this data in some way before any analysis can be successfully employed. Naturally, mining unstructured data is a harder endeavor than mining structured data. Information stored in video, i.e., video and photographs, is also considered to be unstructured data, but its analysis falls out of the scope of this thesis.

There are two main goals that can be distinguished within the area of computer guided data analysis, that of *descriptive* and that of *predictive* data mining. An analyst using a predictive data mining system usually strives for a model that is able to predict certain attributes of a yet unknown "item" based upon some classifier that was constructed through analysis of items that were known at that time. Such a classifier decides on the value of the attribute to be predicted based upon other attributes that are known, usually employing some kind of *divide-and-conquer* principle [14]. Often, such classifiers are represented by decision trees. In contrast with the predictive approach, a descriptive algorithm tries to discover covering knowledge about a certain (application) domain, dealing with commonalities between items or attributes, rather than their distinguishing properties. Such algorithms typically yield correlations that occur often (have a high *support*) within a certain data set that is representative for a certain domain and hopefully have meaning in the real world, e.g., suburban children are often over-weight. The results of descriptive data mining are diverse and have various visualization possibilities, the most common being a clustering, that visually separates certain subgroups based upon some mutual distance, a list of relations between attribute values, that signifies which attributes appear *frequently* together, and a link or network analysis that draws a network of people based upon, for example, a list of phone calls or emails, to identify central figures in a social environment. Given these subdivisions, Figure 1.1 provides an rough impression of the research area of data mining.

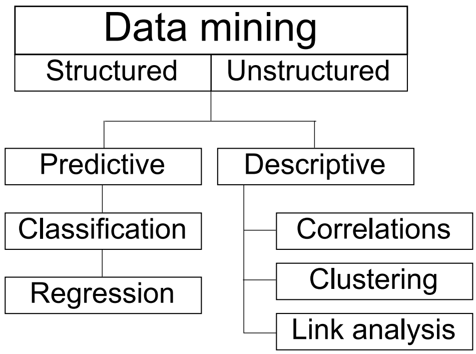


Figure 1.1: Data mining ontology

Most of the different types of data mining have been used throughout this thesis and are therefore relevant to the area of law enforcement to some extent.

1.2 Law Enforcement

The area of law enforcement is very broad in origin and involves all kinds of agencies that assist governing bodies at any level of society to enforce their laws on its populace. Examples of these are among others: military police, religious police, customs, secret police, tax authorities, police, etc. Some of these organizations are subject of law enforcement themselves through “quality control” agencies that check them on internal compliance with the law. However, these agencies are most often placed outside the law enforcement definition and are referred to as “Internal Affairs” or ”Professional Standards”. Also, the justice department and its courts are usually seen as a separate entity in this area.

One division that can be made in this rather large collection of governmental activity is the type of people they deal with; while some of the mentioned organizations concern themselves with all people in a certain society, e.g.: tax authorities, most of the organizations only deal with people who actively break the law, e.g.: police. Therefore, a division can be made between administrative, more passive agencies and investigative, more active agencies, although most administrative organizations also have an investigative branch. This thesis focuses on the investigative agencies or branches.

Within this subsection of law enforcement, another division can be made, also on the level of activity. The most active divisions are occupied with *tactical law enforcement*, the effectuation of agency policies in the direct combat of crime, dealing with crime investigations, forensic research and the preparation of criminal cases through its litigation branch, i.e.: the District Attorney. Part of these organization, however, deal with the formation of such policies, like, investigative focus points, division of agency personnel to specific tasks, etc., based upon knowledge of criminal activity within a legislative entity, These activities are seen as *strategic law enforcement*. This view of law enforcement is shown in in Figure 1.2.

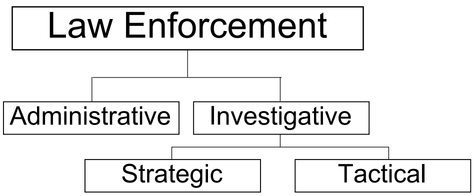


Figure 1.2: The law enforcement area

This thesis is split in two parts, Part I dealing with strategic operations and Part II dealing with the tactical area of policing.

1.2.1 Data on Criminal Activities

Naturally, the agencies mentioned above create a lot of data, exercising their tasks, for example, telephone tapping records, narrative reports, criminal records, and case data. It is clear that most of this data is unstructured, being either recorded (partly) in plain text or not recorded digitally at all. Also all data gathered in the process of law enforcement is governed by very strict preservation and privacy laws. This effectively means, within most legislations, that data can be saved for a very limited time, usually only for the duration of a certain investigation, it can only be used for purposes within that investigation and can only be handled by individuals who are involved in that investigation. Naturally, both issues negatively affect any data mining effort greatly.

Next to that fact, knowledge gained from statistical analysis, of which data mining is a part, has no actual power as evidence in most courts of law, limiting the usefulness of any knowledge from these efforts to the investigation itself, where police officers are still required to gather more "true" evidence.

Also, it is debatable if data mining can play a role as predictive tool in any investigation. Most legislations allow for search warrants and arrests only if certain locations or people can undeniably be linked to a certain crime. The nature of predictive data mining, however, is to transfer knowledge learned from the many to the one, accompanied with a certain reliability percentage, but never in relation to a specific crime. For example, the chance that each of 50 individuals are drug dealers may be 95%, which is considered to be highly reliable, but law enforcers are not allowed to search all these people's homes every time a drug deal goes down. Because of this discrepancy, predictive analysis is not always allowed. leading to "tunnel vision" and illegitimate warrants or arrests. Some of these issues, relevant in the Netherlands, are discussed in [16, 17]. Due to these existing limits on the adoption of data mining in investigative law enforcement, a larger part of all efforts is being directed toward strategic rather than to tactical law enforcement purposes.

1.2.2 Related Work on Data Mining for Law Enforcement

Despite these drawbacks, the number of data mining projects in the law enforcement area is now slowly increasing, both in- and outside of the academic world. Commercial players vary from very small to very large multi-nationals, like statistical software producer SPSS. One of the first large-scale academic projects is the COPLINK project in Arizona where some excellent work has been done in the field of entity extraction from narrative reports [4], the exploitation of data mining for cooperation purposes [3] and social network analysis [20, 5]. In the often mentioned FLINTS project, soft (behavioral) and hard (fingerprints, DNA) forensic evidence was combined to give analysts the ability to build a graphical image of (previously unthought-of) relations between crimes and criminals. Another link-analysis program, FinCEN [7], aimed to reveal money laundering networks by comparing financial transactions. Also, Oatly et al. did some link analysis work on burglary cases in the OVER project [13]. Clustering techniques have also been employed in the law enforcement area. Skillicorn [18] did some work on the detection of clusters within clusters to filter the surplus of information on possible terrorist networks and present law enforcement personnel with a viable subset of suspects to work with. Adderly and Musgrove [1] applied clustering techniques and Self Organizing Maps to model the behavior of sex-offenders. This thesis aims to augment the existing palette of algorithms through the incorporation of new insights into this area.

1.2.3 Motivation

This thesis results from a cooperation between the Dutch National Police (KLPD) and the academic world and the belief that a number of goals set in the area of combating crime can be reached with the assistance of computer guided data analysis, especially where it concerns unstructured or semi-structured data sources. It is believed that fundamental research driven by questions arising from law enforcement practice can lead to a long-term data mining framework centered around the themes knowledge engineering and learning, which is demonstrated by the research described in this thesis.

1.2.4 Overview

This thesis is divided into two parts: one part about algorithms that can be employed for strategic purposes and one part about applications in the tactical domain.

Chapter 2, being the first chapter of the strategic part, describes a first analysis of a large database of criminal records (cf. Appendix B), that aims to relate different crimes to each other or to demographic data, based upon frequent co-occurrence in a record. To accomplish this, the existing and well-known Apriori algorithm was adapted to search for this type of connections in this specific database, incorporating solutions to a variety of problems native to data on criminal activities. This file, that was available in an anonymized version, was also used for more fundamental analyses.

Because this file contains an enormous amount of "raw" data, standard methodologies to visualize data are often not very well suited to this task. Chapter 3 therefore describes a method that optimizes the visualization of relations between criminals in this database by using the domain knowledge of the analyst as a vital part of the clustering phase. Because this expert is able to directly manipulate a physical representation of this data, a high quality of visualization can be reached with a minimum amount of computational effort.

An important concept that can be evaluated using the criminal record database is that of the *criminal career*, which can be seen as a temporally ordered series of crimes committed by an individual throughout his or her life. An ad-hoc method is suggested in Chapter 4 that uses four important factors of such a career to calculate distances between different careers. These distances can then be visualized in a two-dimensional clustering. Chapter 5 proposes a number of enhancements of this method, that are proven to be functional in another domain. Next to that, some methods are discussed that could eventually lead to a prediction of new careers, further examined in Part II.

After the completion of a clustering and classifying system, a search for subcareers that occur often can be performed. An even more noteworthy endeavor is to find specific subcareers that are common in one class and are not in all others, taking the role of defining subcareers that can be used to identify certain classes. These opportunities are researched in Chapter 6, where an existing method for market basket analysis is adapted to suit the demands put forward by the search for common subcareers.

In the second part about tactical applications, the possibilities for predicting criminal careers are discussed in Chapter 7, where a method is described that employs the power of a visualization to create reliable predictions through simple mathematical calculations.

This method is effectuated, expanded en tailored towards criminal data in Chapter 8, where the different variables of this method are tested on the actual data. Under certain conditions, this method can predict criminal careers with a high accuracy.

In Chapter 9, an investigation is described that strives to answer the question if files from confiscated computers from crime scenes can be an indication of which of these scenes are related to the same criminal organizations. For this purpose, a specific distance measure was developed that determines the chance that two computers were owned by the same organization. For this project, text mining software was employed that extracted special entities from computers retrieved from synthetical drugs laboratories.

Chapter 10 describes how “online predators”, child sexual abusers on the internet, can be recognized automatically on social networking sites, like the Dutch “Hyves”. A genetic algorithm was designed that automatically selects groups that show a significant difference between predators and regular users in the amount of under-aged “friends” on their respective profiles. It turns out that in some specific cases this variable can be a strong indicator for danger classification of certain user groups.

This thesis ends in Appendix A with some considerations about statistics, law and privacy that play a pivotal role for everybody using, or intending to use (parts of) our work in the daily practice of police matters. It discusses the applicability, statistical relevance and insightfulness of and reservations to our methods in general and police usage in specific, focusing mostly on the methods in Part II, that deal with tactical policing. As an assurance our methods are viewed in the correct context and our tools are used in a concise way, a deliberation on their possibilities and limitations for use within society is both important and natural.

Bibliography

[1] R. Adderley and P. B. Musgrove. Data mining case study: Modeling the behavior of offenders who commit serious sexual assaults. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'01)*, pages 215–220, 2001.

[2] C.M. Bishop. *Pattern Recognition and Machine Learning*. Springer Science, 2006.

[3] M. Chau, H. Atabakhsh, D. Zeng, and H. Chen. Building an infrastructure for law enforcement information sharing and collaboration: Design issues and challenges. In *Proceedings of The National Conference on Digital Government Research*, 2001.

[4] M. Chau, J. Xu, and H. Chen. Extracting meaningful entities from police narrative reports. In *Proceedings of The National Conference on Digital Government Research*, pages 1–5, 2002.

[5] H. Chen, H. Atabakhsh, T. Petersen, J. Schroeder, T. Buetow, L. Chaboya, C. O'Toole, M. Chau, T. Cushna, D. Casey, and Z. Huang. COPLINK: Visualization for crime analysis. In *Proceedings of The National Conference on Digital Government Research*, pages 1–6, 2003.

[6] M. Dunham. *Data Mining: Introductory and Advanced Topics*. Prentice-Hall, 2003.

[7] H.G. Goldberg and R.W.H. Wong. Restructuring transactional data for link analysis in the FinCEN AI system. In *Papers from the AAAI Fall Symposium*, pages 38–46, 1998.

[8] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning; Data Mining, Inference, and Prediction*. Springer, 2001.

[9] M. Hutter. *Universal Artificial Intelligence; Sequential Decisions Based on Algorithmic Probability*. Springer, 2005.

[10] G.F. Luger. *Artificial Intelligence; Structures and Strategies for Complex Problem Solving*. Pearson Education, 6th edition, 2009.

[11] P. Lymand and H.R. Varian. How much information? Technical report, Berkeley, 2004.

[12] J. Mena. *Homeland Security; Techniques and Technologies*. Charles River Media, 2004.

[13] G.C. Oatley, J. Zeleznikow, and B.W. Ewart. Matching and predicting crimes. In *Proceedings of the Twenty-fourth SGAI International Conference on Knowledge Based Systems and Applications of Artificial Intelligence (SGAI2004)*, pages 19–32, 2004.

[14] J.R. Quinlan. *C4.5 Programs for Machine Learning*. Morgan-Kaufmann, 1993.

[15] S.J. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 2nd edition, 2003.

[16] B. Schermer. *Software Agents, Surveillance, and the Right to Privacy: A Legislative Framework for Agent-enabled Surveillance*. PhD thesis, Leiden University, 2007.

[17] R. Sietsma. *Gegevensverwerking in het kader van de opsporing; toepassing van datamining ten behoeve van de opsporingstaak: Afweging tussen het opsporingsbelang en het recht op privacy*. PhD thesis, Dutch, Leiden University, 2007.

[18] D.B. Skillicorn. Clusters within clusters: SVD and counterterrorism. In *Proceedings of the Workshop on Data Mining for Counter Terrorism and Security*, 2003.

[19] I.H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Elsevier, 2nd edition, 2005.

[20] Y. Xiang, M. Chau, H. Atabakhsh, and H. Chen. Visualizing criminal relationships: Comparison of a hyperbolic tree and a hierarchical list. *Decision Support Systems*, 41(1):69–83, 2005.

Part I

Algorithmic Tools for Strategic Law Enforcement

Chapter 2

Adapting and Visualizing Association Rule Mining Systems for Law Enforcement Purposes

Apart from a list of crimes, criminal records contain diverse demographic characteristics of offenders. This information can be a valuable resource in the constant struggle of assigning the limited police work force to the large number of tasks in law enforcement. For this purpose we try to find relations between crimes and even more important between crimes and demographic data. These relations might give insight into the deployment of officers to certain demographic areas or reveal the linkage of certain crime categories that enable legislative bodies to change policy. The nature of the criminal record database makes it hard to use a standard association detection algorithm, because it encompasses several obviously semantically strongly linked attributes, that pollute the process. We therefore propose a number of techniques, like an attribute ban or a semantic split to improve mining results for this dataset. We also provide a means to include demographically infrequent attributes, like "female", into the comparison. We conclude by showing a way of presenting the resulting trie of frequent patterns to the user, i.e.: the law enforcer.

2.1 Introduction

The notion of relations and their discovery has always been one of the core businesses of law enforcement agencies. In particular, the relations between a crime and individuals, but also the relations between evidence and individuals, e.g., a fingerprint and its owner, or between different offenders, e.g., a mob chief and his hitman, are major focus points of daily police operations. These relations are best characterized as being relations within the tactical field, for they are drawn from and applied in the tactical area of policing. These tactical relations are most often revealed by extensive forensic research or the examination of criminal records.

These records provide, however, also the possibility to reveal existing relations on a strategical level. These relations could be used to describe, and more importantly, prevent crime. This class of relations, found in these records, encompasses relations between crime types, and relations between demographic data and crimes. Revealing these relations enables strategically oriented agencies to develop strategies for the deployment of personnel and other resources.

In this chapter we demonstrate a promising framework for revealing strategic relations for criminal records. In Section 2.2 we explain some of the underlying principles and describe the nature of the criminal record database, to which we specifically suited our efforts. This approach is the main contribution of this chapter and can be found in Section 2.3 and Section 2.4.

2.2 Background

Mining frequent patterns is an area of data mining that aims to discover substructures that occur often in (semi-)structured data. The primary subject of investigation is the most simple structure: itemsets. Much effort from the scientific community has gone into the area of frequent itemset mining, that concerns itself with the discovery of itemsets that are the most frequent in the elements of a specific database. The notion of support (the number of times an itemset is contained in an element of the database) for a single itemset was first introduced by Agrawal et al. [1] in 1993. Since then, many algorithms were proposed, most notably being FP-growth, developed by Han et al. [16], and ECLAT, by Zaki et al. [37].

It might prove rewarding to apply these methods to police data to unravel underlying principles in criminal behavior. For this purpose, the database described in Appendix B seems to be best suited. Its content has been used in a number of descriptive projects [7, 8, 26], that aimed at the exploration of criminal careers (cf. Chapter 4 and Chapter 5) or the impact of its digital nature on privacy legislation.

The nature of the database or, on a larger level, the nature of crime in general, is responsible for a large number of over- or under-present attribute values. The number of males in the database is approximately 80% and almost 90% of the offenders were, not surprisingly, born in the Netherlands. In contrast, the addiction indication is only present for 4% of the entire list. In addition to this discrepancy in attribute values, there is also an inherent correlation between certain attributes that can pollute the outcome of a search. These include (semi-)aggregated attributes, e.g., a very strong relation between age of first and last crime for one-time offenders, and relations that are to be expected logically, like for example the fact that Surinam-born people are often of Surinam-descend.

In essence, the above mentioned algorithms are very well suited to the task of discovering frequent itemsets or relations from a criminal record database; they extract frequent attributes and combine them with other frequent attributes to create frequent itemsets of criminal characteristics. These sets reveal relations between crime types, e.g., a murderer is also likely to steal, and between demographic data and crime types, e.g., a crime outside one's own town is most likely theft. The mentioned methods are however not very well suited for dealing with database characteristics like over- or under-presence, which warrants a refit of these algorithm to facilitate a better extraction of these relations. We propose a number of solutions for this task, fitted into one single approach.

2.3 Approach

For the discovery and exploration of the above mentioned relations we propose a method with five steps. First of all, the standard algorithms for searching frequent itemsets usually rely on databases with boolean attributes, hence we need to transform our database to such a format, discussed in Section 2.3.1. As a first step in our extraction algorithm we then offer the user the possibility of excluding some (range of) attributes, called an *attribute ban* (see Section 2.3.2). The third step enables the analyst to define a *semantic split* within the database, describing the virtual line between two ranges of semantically different attributes (Section 2.3.3).

The actual search for frequent itemsets or relations takes place in the fourth step, described in Section 2.3.4. In this phase of the entire process, a number of methods are used to calculate the importance or significance of a certain itemset. The results of these algorithms are then combined into a single result and the decision is made on the incorporation of this relation into the list of relevant end-results. Finally we propose a way of presenting the discovered relations to the police analyst, in a convenient way for further processing.

The entire approach is shown in Figure 2.1, where the top-down placement of the different substeps describes their chronological order.

2.3.1 Database Refit

In this chapter we use the National HKS database of the KLPD. This database, that can be viewed as a collection of criminal records, is also queried in other chapters and we will therefore refer to Appendix B for a detailed description about its contents, compilation and usage. The methods we employ to detect relations are built around databases with boolean attributes, meaning that a single attribute is either present or not present for a certain record (person). The criminal record database we use (described in Appendix B), naturally, contains no such attributes but instead has both numerical (number of crimes, age of suspect) and nominal data (a criminal is either a one-time offender, intermediate or a “revolving door” criminal).

Numerical attributes are discretized into logical intervals, e.g., a ten year period for age. The algorithm creates a new boolean attribute for each of these intervals, tagging the correct attribute true and setting the others to false.

Nominal attributes are split and a new attribute is created for each category. The category that was selected for a certain individual is set to true.

Note that this leads to a large database, column-wise, that is very sparse; most of the columns are set to false (not present) with only one of the new attributes from the original attribute set to true. Also note that this will automatically lead to an abundance of *strong negative relations*, which are relations that appear remarkably seldom together. For example, “male” and “female” should never both be true for a single individual. Since

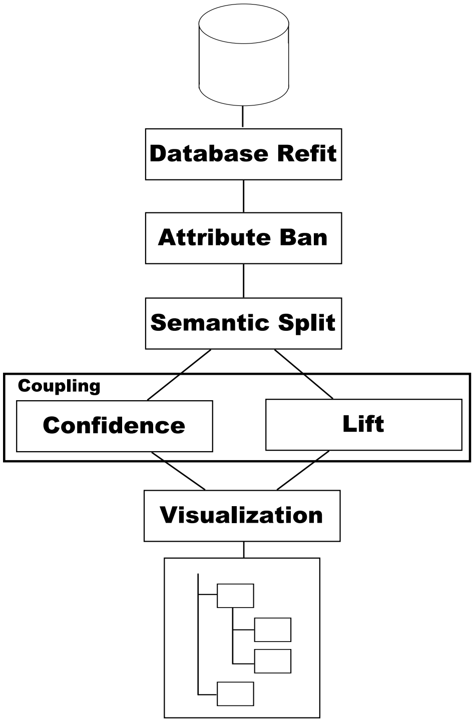


Figure 2.1: Approach for extracting relations from a criminal record database

we are not searching for these kind of relations, this does not pose a problem for the method under consideration. However, a possible extension to this approach, including such search parameters, should be able to deal with this situation (see Section 2.6).

2.3.2 Attribute Ban

In databases there are "disruptive" attributes more often than not. These attributes are on the one hand overpresent in the database while lacking significant descriptive value on the other. One could, for example, consider a plastic shopping bag from a super market. These items are sold a lot and are therefore present in a multitude of *transactions* (or rows) in a sales database. They have therefore a high chance of appearing in a frequent itemset, while their descriptive value is very low; they do not describe a customers shopping behavior, only perhaps that he or she has not brought his or her own shopping bag to the store.

There are a lot of these attributes present in the criminal record database. One of them is, for example, the *deceased* attribute. Since the database itself has only been in use for 10 years, this attribute is most often set to false, leading to an aggregated attribute (see Section 2.3.1) *alive* that is almost always present. The descriptive value of such information to the detection process of relations between criminal behavior characteristics is quite low; the fact whether or not a person is deceased has no relevance to his criminal career or to the presence of other attributes.

To cope with the existence of such attributes in the dataset, we introduce an *attribute ban*; a set $\mathcal{B}$ of attributes or ranges of attributes that are not to be taken into account when searching for relevant relations:

$$\mathcal{B} = \{x \mid x \text{ insignificant attribute}\} \cup \{(y, z) \mid y \leq z, \forall\, q \text{ with } y \leq z, q \text{ insignificant attribute}\},$$

where the attributes are numbered $1, 2, \ldots, n$. Elements can be selected as disruptive and semantically uninteresting by a police analyst, which warrants inclusion into the set during runtime of the algorithm. This set is evaluated in a later step, when the significance of certain itemsets is calculated (see Section 2.3.4).

2.3.3 Semantic Split

A priori knowledge about the semantics of the data under consideration can be a very valuable tool in the data mining process [29]. Especially in the case of the criminal records dataset a clear semantic distinction can be made between the list of crimes on the one hand and demographic data concerning the perpetrator on the other. These two *semantic halves* are strictly separated by the numbering of attributes. In our approach, the data analyst is given the option to either use a *semantic split* by specifying the beginning attribute x of the second halve, or waive this option. From this point on, the algorithm will only combine 1 attribute of one halve (the *lower halve*) with any number of attributes from the other (the *upper halve*). The analyst can define the lower and upper halves by setting either a $1{:}N$ relation (all attributes lower than x are in the lower halve), or a $N{:}1$ relation that sets all elements greater than x as part of the lower halve. Internally, we will mark all the attributes in the lower half by inverting their sign. The semantic split x and the tagging function $\mathcal{S}$ are then defined by:

$$\mathcal{S}_x(y) = \begin{cases} -y & \textit{if } (y < x \textit{ and } 1{:}N) \textit{ or } (y \geq x \textit{ and } N{:}1) \\ y & \text{otherwise} \end{cases}$$

where y is an attribute denoted by a number.

Employing this method, the analyst can use his inside knowledge of the semantics to prohibit a multitude of relations within one semantic halve from appearing into the results. A major example of this occurs within the demographic halve of the database where people from a certain country are most often also born in that country and of that country's ethniticity. In dealing with this situation, police analysts can choose for analyzing the dataset on a $1{:}N$ basis with a semantic split between demographic and criminal data. The semantic split is evaluated during the calculation of significant relations, discussed in Section 2.3.4.

2.3.4 Detection

The actual detection of relations takes place based upon standard frequent itemset mining algorithms. The concept of *support* is the primary unit of comparison used within these techniques. The support of an itemset a ($supp(a)$) is defined as the amount of database records that contain all of the items in the itemset a. Itemsets are considered to be *frequent* when their support is above a certain *threshold*. We define the standard rating based on support for tuples of itemsets a, b as the support of the union of a and b:

$$\mathcal{R}_{\text{standard}}(a, b) = supp(a \cup b)$$

This approach suffices for standard applications, but the above mentioned concerns force our approach to resort to other comparison methods. These methods, described below, where x, y, a and b are itemsets, strive to detect itemset *interestingness* rather than frequency.

Confidence

It might be worthwhile to employ the conditional probability of a certain itemset given another itemset, thereby relinquishing the usage of support. Such a probability, called the *confidence* (of $x \rightarrow y$), is defined by:

$$C(a, b) = \frac{supp(a \cup b)}{supp(a)},$$

when $supp(a) \neq 0$.

When a certain itemset strongly implies another, the combination of itemsets may also be considered interesting. Such a combination has a high confidence for one of the two possible implications. We therefore rate the proposed new itemset on the maximum of both confidences:

$$\mathcal{R}_{\text{both}}(a, b) = max(C(a, b), C(b, a))$$

If both a certain itemset strongly implies another and the other also strongly implies the first (both confidences are high), they can easily be considered to be interesting. Usually, such a set is referred to as a *hyperclique*. If this is the case, the average of both confidences should also be relatively high. The new *candidate* for being an interesting itemset is rated in this way as follows:

$$\mathcal{R}_{\text{avg}}(a, b) = avg(C(a, b), C(b, a))$$

Lift

An itemset will certainly be interesting if its support is much higher than one would expect based upon the support of its individual member-itemsets, the subsets that comprise the itemset. The relation between expected support and actual support is the *lift* of a certain combination of itemsets. We can rate a candidate interesting itemset on this relation calculated by:

$$\mathcal{R}_{\text{lift}}(a, b) = \frac{supp(a, b)}{supp(a) \times supp(b) \ / \ rows},$$

where *rows* is the number of rows (persons) in the dataset.

Combination

For each of the four ratings mentioned above, a different threshold can (and should) be chosen. For the criminal record database, the threshold for $\mathcal{R}_{standard}$ and $\mathcal{R}_{both}$ should be relatively high due to over-presence, while the threshold for $\mathcal{R}_{avg}$ can be relatively low. Combining the four different rating results for a candidate interesting itemset can easily be done by dividing the ratings by their own respective threshold $\mathcal{T}$. The maximum of the resulting percentages will be assigned to the candidate as its score $\mathcal{P}$:

$$\mathcal{P}(a, b) = max\left(\frac{\mathcal{R}_{\text{standard}}(a, b)}{\mathcal{T}_{\text{standard}}}, \frac{\mathcal{R}_{\text{both}}(a, b)}{\mathcal{T}_{\text{both}}}, \frac{\mathcal{R}_{\text{avg}}(a, b)}{\mathcal{T}_{\text{avg}}}, \frac{\mathcal{R}_{\text{lift}}(a, b)}{\mathcal{T}_{\text{lift}}}\right)$$

If this score is higher than 1, one of the thresholds is reached and the candidate itemset is eligible for the so-called notability status.

The search for interesting itemsets (relations) starts with the itemsets of size 1. These itemsets can only be subject tot analysis by $\mathcal{R}_{standard}$, because the other rating systems require at least two itemsets to be compared. For those algorithms, all one-sized itemsets are assumed to be interesting. In the next phase of the algorithm, all itemsets that are considered interesting will be combined with each other to form candidate itemsets. When this step ends we combine the newly found interesting itemsets with all others. This process continues until there are no more new interesting combinations to be found.

Note that the semantic split and attribute ban are also taken into account when single attributes are selected to form a new candidate itemset resulting in Algorithm 1. The product of the elements of an itemset x will be denoted by $\mathcal{I}$:

$$\mathcal{I}(x) = \prod_{i \in x} i$$

This algorithm employs and yields a trie, an ordered tree data structure that is used to store an associative array, and that facilitates efficient storage and easy retrieval of interesting relations.

It may be the the case that interesting itemsets of size larger than 2 exist, where none of its children is considered to be interesting. These itemsets will not be detected by the current version of our approach because of the iterative way the itemsets are constructed and not all of the calculations adhere to the Apriori property, that states that an itemset can only be frequent if all its sub sets are also frequent. Although these itemsets are believed to be very uncommon, the results of our approach should be viewed as a good first approximation of the complete result set.

var include := true
var include2
do
  foreach interesting set x with $size(x) = 1$
    if $x \in \mathcal{B}$ **then** include := false
    foreach interesting set y with $size(y) > 1$
      if $x < 0$ **and** $\mathcal{I}(y) < 0$ **then** include2 := false **else** include2 := include
      if include2 = true **and** $\mathcal{P}(x, y) > 1$ **then** (x, y) is interesting
    endfor
  endfor
until no more new interesting itemsets

Algorithm 1: The approach for finding interesting relations

2.4 Trie Visualization

It is obviously important to produce the end results in such a way to the police analyst that he or she can immediately understand and interpret them. For this purpose we need to find a scalable (the trie is large) and fitting metaphor to describe our trie, that a non-computer scientist can easily relate to [15].

One of the few tree-related metaphors common computer users are familiar with is that of the directory or folder structure on a harddrive and more specifically the Microsoft Windows folder browse control, displayed in Figure 2.2. We propose to use this metaphor to browse the trie.

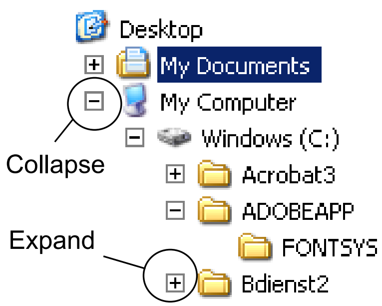


Figure 2.2: A standard Windows folder browse control

In this control, the directory structure is built up from its root, which is at the start the only node made visible to the user. If a certain node has any children, a plus sign (the *expand button*) is put next to it, which, when clicked upon, “reveals” the children of this node by showing them, indented, underneath their parent directory. After such an operation, the expand button changes into a minus sign, which, when clicked, *collapses* the node and hides the entire subtree under its assigned node. The familiarity of this tree representation helps the police analyst in easy exploration of the data through a simple clicking interface, where each node is (part of) a discovered relation and is accompanied by its threshold reaching score.

The most important feature of this visualization is however the scalability of its content. Tries resulting from a frequent pattern search can be quite large, which makes it hard to easily browse the outcome of one's search, especially when itemsets contain more than two items. Hiding subtrees behind expand buttons enables the police analyst to limit the examination of relations to the ones that appear to be interesting, just by glancing at the first element of the itemset.

For this convenient selection of possibly interesting relations, the efficient trie needs to be transformed to its full lattice on the screen, thus making sure that each attribute that is part of *any* relation will be displayed in the root list of our control. If any of these attributes arouses the interest of the analyst, examination of this relation can start by a single click on its expand button. An example of how our method produces the end results on screen can be seen in Figure 2.3.

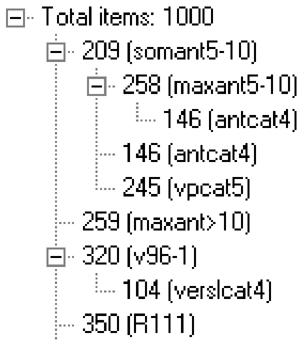


Figure 2.3: Results of a certain investigation

2.5 Experimental Results

We tested our algorithm and visualization tool on the database in Appendix B, containing approximately one million offenders and the crimes they committed. Our tool analyzed the entire database of criminals and presented the user with the resulting relations between criminal characteristics.

Some of the most notable relations have been made available to police experts in the field of strategic analysis and can contribute to policy updates in the fight against crime. Most of them were reached within a setting of either searching between crimes alone (banning all attributes in the demographic halve) or when employing a semantic split with a 1:N relation between demographic data and the list of crimes. The other settings used in the experiments resulted into a list of relations that contains much jitter. Because a number of customizable settings is available, it is to be expected that the future will reveal a number of other option sets that give good results, especially after the tool has been incorporated into everyday use by the experts. Below we show some of the most remarkable and recognizable results from our experiments. The first set describes relations between police data, the second set contains demographic data as well.

Joyriding ↔ Violation of Work Circumstances ↔ Alcohol Addiction
Drug Smuggling ↔ Drug Addiction
Manslaughter ↔ Discrimination

Male ↔ Theft with Violence ↔ Possession (of weapon)
Female ↔ Drug Abuse
African Descend↔ Public Safety
Rural Areas ↔ Traffic Felonies

The confidential nature of the data used for this analysis prevents us from disclosing more detailed experimental results reached in our research.

2.6 Conclusion and Future Directions

In this chapter we demonstrated the applicability of frequent itemset mining in the analysis of criminal characteristics for strategic purposes. The tool we described compiled a list of noteworthy relations between crime types and most important demographic characteristics. The nature of a criminal record database established the need for specifically suited adaptations of standard mining algorithms to cope with over- and under-presence of and inherit relations between attributes. The end report consists of a visual, scalable and clickable version of the resulting trie and is ready to be used by police experts.

The semantic split proposed in this chapter already exploits the semantic knowledge of the analyst using the system. This can be extended to a more detailed level, a *semantic bond*, where semantic overlaps between two or more attributes can be defined. Characteristics in such a set should then not be combined in the detection phase. This way the coarse semantic split can be updated to a finer level of semantic coherence.

For this research, the search for relations was focused on positive relations, meaning, that two or more attributes appear notably often together. It may also be of interest to the law enforcer to search for attributes that appear reasonably seldom together. However, the search for those relations with our method is hindered by the boolean nature of the database, required by standard approaches, and the way we aggregate those from the original nominal or numerical attributes: aggregated attributes never appear together by definition. One way to solve this might be to join them into a semantic bond as mentioned above. Other possibilities might also be applicable.

Future research will aim at improving on the concerns mentioned above. After these issues have been properly addressed, research will mainly focus on the automatic comparison between the results provided by our tool and the results social studies reached on the same subject, in the hope that "the best of both worlds" will reach even better analyzing possibilities for the police experts in the field. Incorporation of this tool in a data mining framework for automatic police analysis of their data sources is also a future topic of interest.

Chapter 3

Object-Centered Interactive Multi-Dimensional Scaling: Ask the Expert

Multi-Dimensional Scaling (MDS) is a widely used technique to show, relations between objects—such as humans, documents, soil samples—that are defined by a large set of features, in a low-dimensional space. Key benefit is that it enables visual inspection of object relations in an intuitive, non-technical way, very well suited for, for example, police officers without technical backgrounds. One of the limitations is that different projections exist, leading to different graphical representations and therefore different interpretations of the data. This problem is made more significant in case of noisy data or heuristic approaches to MDS. We propose Object-Centered Interactive Multi-Dimensional Scaling (OCI-MDS), a technique that allows a data expert, for example a police analyst, to try alternative positions for objects by moving them around the space in real time. The expert is helped by several types of visual feedback, such as the proportional error contribution of the expert-controlled object. Here we show that this technique has potential in a number of different domains.

3.1 Introduction

The use of computers has enabled people to create large amounts of data. This is a trend that is not restricted to a specific domain. For example, policy makers write large numbers of reports, individuals publish personal web-pages, police officers create large numbers of case data, and databases enable structured storage of large amounts of medical data. Extracting potential relations between data objects—i.e., an individual data element such as a document—is a current challenge. Many techniques have been developed for this purpose, like data clustering, graph-mining and Principal Component Analysis (PCA) [24]. In this chapter we focus on one such technique, Multi-Dimensional Scaling [13, 31].

Multi-Dimensional Scaling (MDS) is a widely used technique to show, in a low-dimensional space, relations between objects—such as human subjects, documents, soil samples—that are defined in a higher-dimensional space. If MDS is used to create a 2D visual representation of the high-dimensional dataset, a key benefit of this technique is that it enables visual inspection of object relations in an intuitive way. This is important, especially when the users of the analysis (i.e., those interpreting the final 2D projection) are not machine-learning experts. One of its limitations, however, is that different projections exist, leading to different graphical representations and therefore different interpretations of the data. This problem is especially important in case of noisy data or heuristic approaches to MDS. First, noisy (or unstructured) data introduce variation in the high-dimensional distance between objects, and as such these variations will be reflected in the 2D projection. As this noise does not convey any information regarding the relation between objects, interpretation of the 2D projection is hindered by this noise. The algorithm does not know the underlying relation between objects, and as such cannot correct for it. An expert could. Second, heuristic approaches to MDS, such as the push-pull technique [11, 20] where the projection is constructed through repeated local comparison between pairs of objects, introduce sub optimality in the 2D projection and can converge to local minima. However, heuristic approaches can have important benefits such as reduced computational cost and scalability [35] and are therefore useful for solving MDS problems.

In this chapter we propose Object-Centered Interactive Multi-Dimensional Scaling (OCI-MDS), a technique that allows a data expert to propose alternative positions for objects by moving them around the 2D space in real time. The approach is compatible with (and helps) heuristic approaches to MDS. The expert is helped by several types of visual feedback, such as the proportional error contribution of the controlled object. We use the technique in a heuristic MDS approach and show that this technique has potential in two different domains: visualization of high-dimensional computer simulation experiment configurations [6] and raw biomedical data.

Our interactive approach relates to other approaches, such as those by Stappers et al. [28]. They use interaction to enable exploration of data. Objects can be selected by the user, after which the algorithm clusters the newly selected object. Subsequently, a next object can be added (or removed). This approach is Object-Centered and allows expert-controlled visualization of object-relations, but different in the sense that objects, once positioned, are not interactively movable to (a) generate alternative hypotheses about object relations, or (b) help the MDS mechanism. Further they focus on small amounts of objects (about 10). Other approaches include non Object-Centered ones, such as those that enable experts to direct computational resources at specific parts of the space in order to reduce computational resources needed for data projection [35], and those that enable experts to interactively change algorithm parameters (like noise) and to stop the algorithm [30].

In Section 3.2 we describe some of the problems our approach addresses. In Section 3.3 we introduce Object-Centered Interactive MDS. Section 3.4 presents experimental results and contains all figures. Finally, we present our conclusion and some directions for future work in Section 3.5.

3.2 Expert Interaction Helps Heuristic Projection

A key motivation to use MDS for visualization of high-dimensional data is its ability to give an overview of a complete dataset. This is important in the exploratory phase of data analysis. For example, in the criminal investigation area, visualization of datasets supports police officials in their process of generating hypotheses about the relation between different criminal records [11]. In the computer simulation domain, such as simulation of adaptive behavior [6], scientists often repeat experiments with slightly different settings. It is important to avoid making errors in the configuration of the experiments and it is important to have a clear overview of the variations introduced in the parameters. Visualization of the relation between configurations of these experiments (not just the results) can therefore provide insight into both the completeness of a set of experiments as well as potential configuration mistakes. In the domain of biomedical data analysis, clustering, dimension reduction and visualization are used to, for example, find in a set of patients different variations of one disease, or find the most important factors underlying a certain disease.

In all those domains, MDS can be used to cluster high-dimensional data by projecting it onto a 2D space (note that data is not really clustered, as explicit groups are not made). Visualization of that space enables domain experts to get an intuitive idea of the relation between the objects in high-dimensional space. Typically, a 2D projection is constructed such that the least-square error is minimized (see Section 3.3). However, an error of 0 is usually not possible, and, if the projection technique is heuristic, minimal error cannot be guaranteed.

Another typical problem in heuristic approaches—that use incremental error minimization by inducing small changes to object locations in 2D—is that two objects that should be close to each other can be separated by a large cluster, because the large cluster pushes both objects away from each other (see Figure 2, Section 3.4). Standard incremental techniques cannot solve this problem. Thus, even though the solution is near optimal, large local errors can exist.

However, domain experts can detect such local errors by looking at the object and comparing it with its neighbors. So, from an optimality point of view the ability to move objects and clusters of objects is a useful addition to heuristic approaches to MDS. For data interpretation it is also a useful addition, as interactive real-time movement of objects enables experts to test hypotheses of relations between objects directly in the clustering result. This means that, e.g., police officials are able to test if two criminal records are related just by moving the objects close to each other and observing, e.g., the clustering result. Another advantage is the possibility to add new objects at user specified locations, and observe the trajectory of these objects in the projection as well as the influence of these objects on the projected location of other objects.

To summarize, object-based interaction with MDS is useful, provided that users get feedback information so that they can (1) select objects to move, and (2) evaluate the result of the move.

3.3 Object-Centered Interactive MDS

We propose Object-Centered Interactive MDS (OCI-MDS). This allows experts to interactively manipulate the projection result produced by a heuristic MDS algorithm. We present our algorithm and the kind of user-feedback the system gives. In the next section we show that it is a useful technique in two domains: computer simulation experiments and biomedical data analysis. Analogous to standard MDS, four steps are needed to project m-dimensional data onto a low-dimensional (2D) space. The first two are preparatory and the second two are iterative until some stop-criterion (usually reaching a minimal error, or stalled improvement for some fixed number of iterations).

First, a distance matrix is constructed that captures the distances between n individual objects in m-dimensional space, where m typically is the number of features used to describe an object. If objects do not have the same number of features (i.e., objects in one set have different dimensionality) then the distance matrix must be able to cope with this. We assume we have such an $n \times n$ distance matrix D.

Second, objects are randomly placed in a low-dimensional space, in our case a 2D space. A vector O of size n represents the coordinates of all n objects.

Third, the first iterative step selects (e.g., randomly) an object i and adds random noise to the coordinates of that object. Noise is added as follows:

$$O_i[x] \leftarrow O_i[x] + \mathrm{rnd}() \cdot step$$

$$O_i[y] \leftarrow O_i[y] + \mathrm{rnd}() \cdot step$$

where $O_i[x]$ and $O_i[y]$ are the coordinates of an individual object i, and rnd() a function giving a random number in $[-0.5, 0.5]$. The variable *step* is a noise size factor that is local-error, total 2D space span, and annealing dependent:

$$step = \alpha \cdot d \, \frac{n \cdot e_i}{e}$$

where d is the largest distance between objects in O, and thus equivalent to $\max(D^L)$ (see below), e_i the local error associated with object i, e the global error (see below), and α an exponentially decreasing annealing factor. The local error e_i is defined by:

$$e_i = \sum_{j=1}^{n} (D_{ij} - D_{ij}^L)^2$$

where D^L is the distance matrix of objects in 2D space. The motivation for our step factor is as follows. It expresses a normalized maximum step that depends on the error contribution of an object to the global error and the size of the space covered by all objects. This has the following two benefits. First, an object with a high local error is moved through the 2D space with high speed, in order to find a better location for it. This increases the probability that wrongly placed objects eventually find a suitable location with small local error. Second, if the 2D space that is occupied by the objects is large, the objects will also move quicker. This ensures that the algorithm is not dependent on the absolute distance between objects. Further, we decrease the annealing factor α exponentially whenever for all objects i there has been no decrease in e. So, if the algorithm approaches a minimum, smaller steps can be used to better approach that minimum.

Fourth, update the distance matrix D^L (note that we use Euclidean distances for D^L). Then evaluate the least-square error (LSE) between D and D^L:

$$e = \sum_{i=1}^{n} \sum_{j=1}^{n} (D_{ij} - D_{ij}^L)^2$$

If the local noise added to object i decreases global error e, keep the new coordinates; if not, discard the change. Repeat step three and four until e is smaller than a threshold t, or until e has not decreased for a fixed number of steps s. If this criterion is met, the process is paused until the user interactively changes positions of objects.

Objects are drawn in a two-dimensional plane (e.g., Figure 3.1(a), Section 3.4). The user is able to, at any time, grab objects and place them at alternative positions. This enables the user to (1) help the heuristic MDS, and (2) experiment with potential clusters. The user is given two types of direct feedback. First, when objects are released, the projection mechanism restarts iteration of step three and four, so the user can directly observe the effect of the moved objects on the total distribution of objects. Second, objects are drawn in a color that represents the local error contribution of the object. This is non-trivial, as color changes need to be reactive enough to reflect small changes in local error but also reflect global changes in global error e. We used the following formula:

$$color_i = \log\left(1 + \frac{n \cdot e_i}{\log(1+n) \cdot e_{min}}\right)$$

where n is the number of objects, e_i the local error of object i and $e_{min} = \min(e_s)$ for $s = 0, 1, 2, \ldots$, where e_s is the global error at iteration s. The variable $color_i$ can be used to define, for example, drawing intensity or color of an object i. This color scheme was used to experiment with interactive visualization of simulation experiment configurations, as well as interactive visualization of biomedical data.

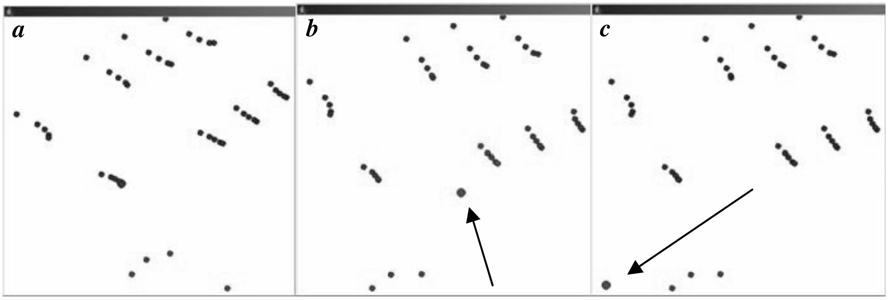
We also experimented with a different coloring scheme where objects are colored using a per object measure that is relevant to the dataset, not the algorithm. This scheme was used to experiment with biomedical data visualization. The data consisted of about 400 objects with 10 features. The objects are patients. Every feature represents the severity of a disease in a different part of the body. The color represents the total severity calculated by averaging over the different features. This average is meaningful but somewhat arbitrary, as it is not clear that the average is actually a good representation of the global severity of the disease. However, as our focus is the visualization technique, not the dataset, we do not consider this to be a problem at this moment. For the same reason we do not specify the datasets in detail in this chapter.

3.4 Experimental Results

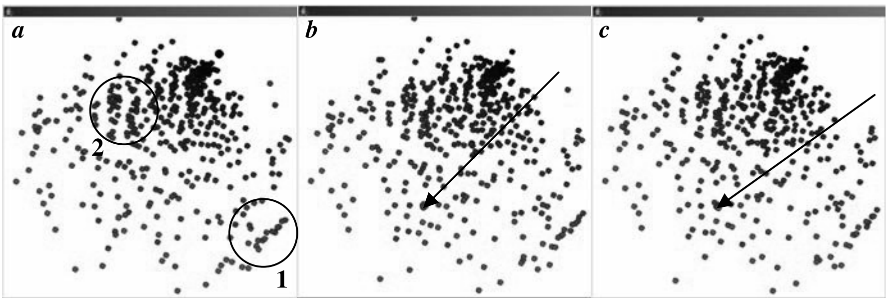
We have developed a Java application that allows us to test our approach. First we present results that investigated its use in visualizing simulation experiment configurations. The dataset consisted of 44 experimental configurations, all of which are used in research into the influence of emotion on learning using reinforcement learning [6]. The features of the objects consisted of 40 different learning parameter settings such as learning rate, exploration-exploitation settings, etc. This can be considered structured data. We have a vector representation of these configuration documents, and defined a distance measure based on the parameter type (boolean, string, double) and the range of values found for one parameter. We do not detail the distance measure here. Based on the measure we constructed the distance matrix D.

Figure 3.1(a)a shows an initial 2D projection of a set of experiment configurations. The visualization clearly shows that there are 4 experiments that are special (bottom), and several groups of other experiments. The objects at the bottom are control experiments, and are indeed the control experiments with which the others are compared. The control experiment at the right is further away from the other three (and further away from all other experiments). Did our algorithm correctly place it here? The user can grab (Figure 3.1(a)b) the object, and while moving it, the local errors start to increase (objects color red). Apparently, the object should not be moved in that direction. After letting the object go, the algorithm projects the object back to a similar (but not necessarily equal) position. The object indeed belongs there. Other object clusters show a nice regular pattern, as a result of the distance function. The four top clusters (Figure 3.1(a)c) all belong to one typical parameter value, while the middle four all belong to a different value on that same parameter. The clusters themselves are organized and correspond well to the actual configurations. This enabled us to verify that no configuration errors had been made in the course of these experiments.

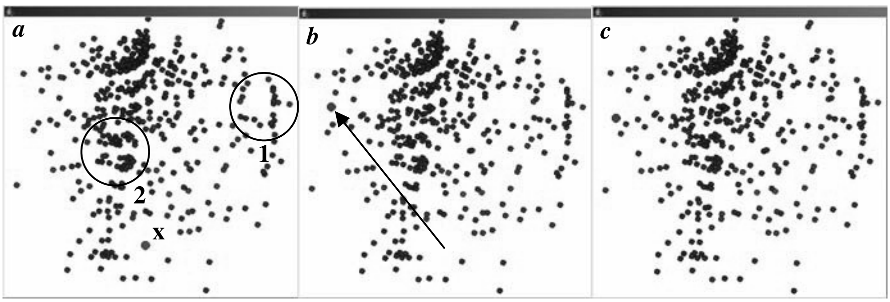
Second, we experimented with the biomedical data mentioned earlier. The projection resulted in a clustering that showed a trend from high severity to low severity, even though global severity is not a feature in the data (Figure 3.1(b)a). Although the projection clearly does not give as much insight into the data as the projection of the structured experiment data shown before, several clusters appear to exist. For example, cluster 1 represents a coupling of two severely affected body parts. Cluster 2 represents a coupling of two other severely affected body parts where the two parts of cluster 1 are not affected. This might indicate correlation between certain body parts and dissociation between others. Although the heuristic technique extracts some useful information from unstructured raw biomedical data, as it stands, the technique is clearly not usable for quantitative conclusions about the dataset, but only for explorative data analysis. However, the main topic of this chapter is dataset exploration and interactivity. Interaction enabled us to relocate two items that appeared to be badly clustered due to the separation problem mentioned earlier, i.e., a cluster divides otherwise closely related objects (Figure 3.1(b)b and c). After grabbing and relocating the two severe disease cases to an arbitrary position on the other side of the cluster consisting of non-severe disease objects, they were relocated by the algorithm at positions that better matched their real-world relation, as could be observed from a comparison with the objects near to that new location. Finally, Figure 3 shows the variation in the distribution of local errors. Figure 3.1(c)a also shows one object ('x') with high local error positioned in between objects with small local errors. When grabbing and repositioning the object at a location in which it appears to have smaller local error, we were able to relocate it at a better place. Although the exact meaning of the projection is at this point unclear (and strongly dependent on the distance measure we used), our experiment shows that Object-Centered interactivity is a useful method to explore object relations.



(a) Manipulating experiment configuration clusters (local error color)



(b) Manipulating biomedical data (severity color)



(c) Manipulating biomedical data (local error color)

Figure 3.1: Experimental Results

3.5 Conclusion and Future Directions

Our experiments show that Object-Centered Interactive MDS has potential. It can be used for direct manipulation of clustering results based on a heuristic MDS approximation. It can help in verifying the MDS result, and help to generate hypotheses about alternative object relations, that were not found, for example, because the MDS converged to a local optimum. However, currently its usefulness is somewhat limited on highly unstructured data.

Future work includes adding multiple-object drag-and-drop, extensive user testing, and scaling mechanisms like those introduced by Williams and Munzner [35]. Also, relating the re-placement of single items with a high local error (a high contribution to the global error) to the change in the global error is important for the analysis of the proposed approach. Changes in the global error can be represented by a Boolean figure (higher or lower) or be represented by a (color-)scale during the process of human intervention. Future research can strive to find a relation between the decrease in local errors and the error made in the total image. If such a positive relation exists, automating the process of relocating those items with the highest local error can be an asset worth pursuing. This global optimization can be of interest in areas where the correctness of the image as a whole is more important than the relation between a small subset of individual items, like, for example, a clustering on customer behavior.

Our main goal is to use the proposed tool as part of a larger data mining framework for law enforcement purposes as described in Chapters 4 and 5. These chapters focus on the contributions a domain expert can make to the process of analyzing criminal careers using our interactive clustering tool.

Chapter 4

Data Mining Approaches to Criminal Career Analysis

Narrative reports and criminal records are stored digitally across individual police departments, enabling the collection of this data to compile a nation-wide database of criminals and the crimes they committed. This collection of data over a number of years presents new possibilities of analyzing criminal activity through time. Augmenting the traditional, more socially oriented, approach of behavioral study of these criminals and traditional statistics, data mining methods like clustering and prediction enable police forces to get a clearer picture of criminal careers. This allows officers to recognize crucial spots in changing criminal behavior and deploy resources to prevent these careers from unfolding.

Four important factors play a role in the analysis of criminal careers: crime nature, frequency, duration and severity. We describe a tool that extracts these from the police database and creates digital profiles for all offenders. It compares all individuals on the basis of these profiles by a new distance measure and clusters them accordingly. This method yields a visualization of the criminal careers and enables the identification of classes of criminals. The proposed method allows for several user-defined parameters.

4.1 Introduction

The amount of data being produced in modern society is growing at an accelerating pace. New problems and possibilities constantly arise from this so-called data explosion. One of the areas where information plays an important role is that of law enforcement. Obviously, the amount of criminal data gives rise to many problems in areas like data storage, data warehousing, data analysis and privacy. Already, numerous technological efforts are underway to gain insights into this information and to extract knowledge from it.

This chapter discusses a new tool that attempts to gain insights into criminal careers:

the criminal activities that a single individual exhibits throughout his or her life. From the criminal record database described in Appendix B, our tool extracts four important factors (Section 4.2) in criminal careers and establishes a clear picture on the different existing types of criminal careers by clustering. All four factors need to be taken into account and their relative relations need to be established in order to reach a reliable descriptive image. To this end we propose a way of representing the criminal profile of an individual in a single year. We then employ a specifically designed distance measure to combine this profile with the number of crimes committed and the crime severity, and compare all possible couples of criminals. When this data has been compared throughout the available years we use the human centered visualization tool, described in Chapter 3, to represent the outcome to the police analysts. We discuss the difficulties in career-comparison and the specific distance measure we designed to cope with this kind of information.

In Section 4.3 we describe our approach to this problem. The main contribution of this chapter is in Section 4.4 and 4.5, where the criminal profiles are established and the distance measure is introduced. A prototype test case of this research was described earlier in [8].

4.2 Background

In Chapter 2, [10] we made an attempt at revealing what criminal characteristics were linked together through common co-occurrence. The research in this chapter aims to apply visualization of multi-dimensional data to criminal careers (rather than single crimes or crimes to demographic data) in order to constitute a representation of clusters or classes of these criminals.

Criminal careers have always been modeled through the observation of specific groups of criminals. A more individually oriented approach was suggested by Blumstein et al. [4]: little definitive knowledge had been developed that could be applied to prevent crime or to develop efficient policies for reacting to crime until the development of the criminal career paradigm. A criminal career is the characterization of a longitudinal sequence of crimes committed by an individual offender. Participation in criminal activities is obviously restricted to a subset of the population, but by focusing on the subset of citizens who do become offenders, they looked at the *frequency*, *seriousness* and *duration* of their careers. We also focus our criminal career information on the *nature* of such a career and employ data mining techniques to feed this information back to police analysts and criminologists.

4.3 Approach

Our "criminal career analyzer" (see Figure 4.1) is a multi-phase process that works on a criminal record database (in our case, the database described in Appendix B). From these files our tool extracts the four factors, mentioned in Section 4.2, and establishes the criminal profile per offender. We then compare all individuals in the database and calculate a distance based upon this profile, crime severity and number of crimes. This information is summed over time and clustered using a human centered multi-dimensional visualization approach. Black boxed, our paradigm reads in the database and provides a visual career comparison report to the end user.

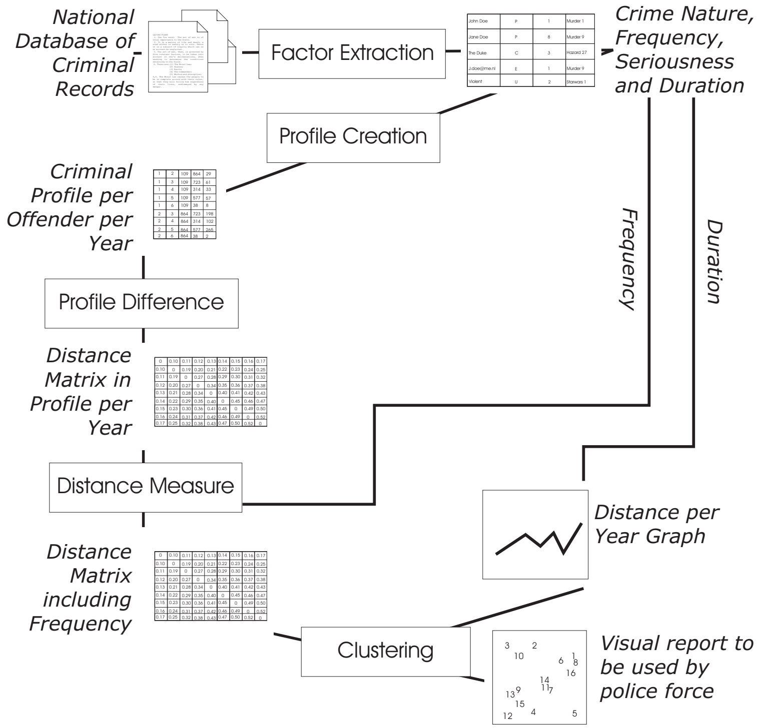


Figure 4.1: Analysis of Criminal Careers

First, our tool normalizes all careers to “start” on the same point in time, in order to better compare them. Second, we assign a profile to each individual. This is done for each year in the database. After this step, that is described in more detail in Section 4.4, we compare all possible pairs of offenders on their profiles for one year, while taking into account that seriousness is inherently linked to certain profiles. The resulting distance matrix is based upon only two of the four before mentioned criteria, crime nature and severity. We then employ a distance measure described in Section 4.5 to fairly incorporate the frequency of delinquent behavior into our matrix which is then summed over the available years. In the last step we visualize our distance matrix in a two-dimensional image using the method described in [5].

The database is an up-to-date source of information and therefore represents the current situation. This does, however, present us with the following problem: it also contains the criminal activities of people that started their career near the end of the database’s range. Our approach suffers from a lack of information on these offenders and when translated, it could be falsely assumed that their criminal activities terminated — however, they have just started. To cope with this problem we draw a semi-arbitrary line after two years, which means we dot not take criminals into account that have only been active for two years or less. In a later stage, it is presumably possible to predict in which direction the criminal careers of these offenders will unfold. More information about this prediction can be found in Section 4.7.

4.4 Obtaining Profile Differences

Interpreting the types of crimes in the database and compiling them into a criminal profile for individual offenders is done both by determining the percentage of crimes in one year that fall into each one of the categories and dealing with these crimes' seriousness. The categories are then ordered on severity. Note that while all categories are considered to be different when determining profile, their respective severity can be the same. We distinguish three types of crime seriousness: minor crimes, intermediate crimes and severe crimes. Each crime type falls within one of these categories. A typical profile of a person looks like the profile described in Table 4.1. Summing all the values in a row in the table will result in 1, representing 100% of the crimes across the categories.

Table 4.1: A typical Criminal Profile

Crime Type	*Traffic Crimes*	*Financial Crimes*	...	*Sex Crimes*
Severity	minor	minor	...	severe
Percentage	0	0.6	...	0.1

Now that we have established a way of representing the crime profile of an offender for a single year, it is possible to compare all individuals based upon this information and compile the profile distance matrix for each year. It is imperative that we take both the severity and nature of each profile into account.

We employ the method described in Table 4.2 to accomplish this. Between all couples of offenders we calculate the *Profile Distance* as follows. First we take the absolute difference between the respective percentages in all categories and sum them to gain the difference between these two offenders in crime nature. This number varies between 0, if the nature of their crimes was exactly the same for the considered year, and 2 where both persons did not commit any crime in the same category. An example of this can be seen in the top panel of Table 4.2, with 0.6 as difference.

One of the problems that arise in this approach is how it should deal with people with no offenses in the considered year. In [8], it was assumed that setting all the percentages in the crime nature table to 0 would provide a result that was true to reality. This does not seem to be the case; such an assumption would provide smaller distances between offenders and "innocent" people, than between different kinds of criminals for an arbitrary year. It would be desirable to assign a maximal distance in crime nature between

Table 4.2: Calculating the difference between Individual profiles

Crime Nature Difference

Ind. 1	0.1	0.0	0.0	0.0	0.5	0.4	0.0	0.0	
Ind. 2	0.0	0.2	0.1	0.0	0.4	0.3	0.0	0.0	Summed
Diff.	0.1	0.2	0.1	0.0	0.1	0.1	0.0	0.0	***0.6***

Crime Severity Difference

	Minor		Intermediate		Severe		
	#	Fac.	#	Fac.	#	Fac.	Summed
Ind. 1	0.1	1	0.5	2	0.4	3	
		0.1		*1.0*		*1.2*	*2.3*
Ind. 2	0.3	1	0.4	2	0.3	3	
		0.3		*0.8*		*0.9*	*2.0*
Diff.							***0.3***

Total Profile Difference: 0.9

innocence and guilt. Hence we propose to add to this table the *innocence column*. An individual without offenses in the year under consideration would have a 1 in this field, while an actual offender will always have committed 0% of his or her crimes of the innocent nature. This ensures that the algorithm will assign the desired maximal distance between these different types of persons.

Taking the severity of crimes into account is somewhat more complex. Not only the difference within each severity class needs to be calculated, but also the difference between classes. For example, an offender committing only minor crimes has a less similar career to a severe crime offender, than a perpetrator of the intermediate class would have to the same severe criminal. In compliance with this fact, our approach introduces a weighting factor for the three different categories. Minor offenses get a weight of 1, intermediate crimes get multiplied by 2 and the most extreme cases are multiplied by 3. Of course, these values are somewhat arbitrary, but the fact that they are linear establishes a reasonably representative and understandable algorithm. This multiplication is done for each individual and is summed over all severity classes before the difference between couples is evaluated. Obviously this will yield a greater distance between a small time criminal and a murderer ($|1 \cdot 1 - 3 \cdot 1| = 2$, which is also the maximum) than between an intermediate perpetrator and that same killer ($|2 \cdot 1 - 3 \cdot 1| = 1$). Naturally, two individuals with the same behavior, still exhibit a distance of 0. An example of this calculation can be found in the bottom panel of Table 4.2, with value 0.3.

Naturally, the innocence category in crime nature will be assigned a severity factor of

0, stating that a minor criminal (for example: theft) will be more similar to an innocent person than a murderer would be, when looking at crime severity.

Both these concepts will ultimately culminate into the following formula:

$$
\begin{aligned}
PD_{xy} \quad = & \quad \left(\sum_{i=1}^{9} \left| Perc_{ix} - Perc_{iy} \right| \right) + \\
& \quad \left(\left| \sum_{j=1}^{4} Fact_j \cdot Sev_{jx} - \sum_{k=1}^{4} Fact_k \cdot Sev_{ky} \right| \right)
\end{aligned}
$$

where the *profile distance per year* between person x and y is denoted as PD_{xy}, the percentage of person x in crime category a is described as $Perc_{ax}$, and Sev_{bx} points to the severity class b of person x, with attached multiplication factor $Fact_b$. This formula yields a maximum of 5 (the maxima of both concepts summed) and naturally a minimum of 0. In the example from Table 4.2 we would get 0.6 + 0.3 = 0.9.

As an example of our approach we look at the following situation for an arbitrary year:

Table 4.3: Four people's behavior in one year

Person	*No Crime*	*Bicycle Theft*	...	*Murder*
Innocent Person	1.0	0.0	...	0.0
Bicycle Thief	0.0	1.0	...	0.0
Murderer	0.0	0.0	...	1.0
Combined	0.0	0.5	...	0.5

First we calculate the difference between all four offenders in both crime nature and crime severity. This yields the following table.

Table 4.4: Crime Nature and Severity Distances

Nature						*Severity*				
	I	*B*	*M*	*C*			*I*	*B*	*M*	*C*
I		2	2	2		I		1	3	2
B			2	1		B			2	1
M				1		M				1

We can now calculate the Profile Difference between the offenders. The result can be seen in Figure 4.2.

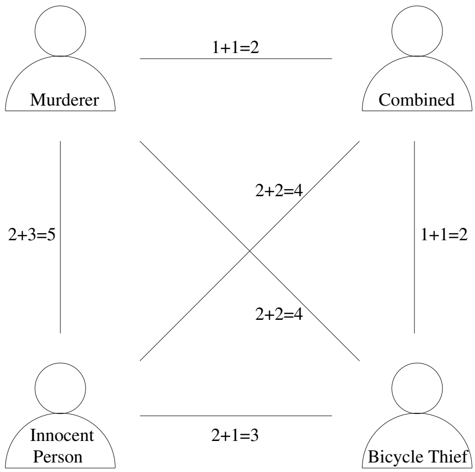


Figure 4.2: Example of Profile Difference between different criminals

We clearly observe from this figure that the innocent person from our list has a large distance from all offenders, varying from 3 to the bicycle thief to 5 to the murderer. As one would expect, the distances between respectively the thief and the thief-murderer, and the murderer and the thief-murderer are rather small.

The intermediate result we have calculated describes the difference in profile and crime seriousness between all individuals in the database. This distance matrix will be used as input for our main distance measure that will add crime frequency to the equation.

4.5 A New Distance Measure

The values the *PD* can get assigned are clearly bounded by 0 and 5. This is, however, not the case with the crime frequency of the number of crimes per year. This can vary from 0 to an undefined but possibly very large number of offenses in a single year. To get a controlled range of possible distances between individual careers it is desirable to discretize this range into smaller categories. Intuitively, beyond a certain number of crimes, the difference that one single crime makes is almost insignificant. Therefore we propose to divide the number of crimes into categories. These categories will divide the crime frequency space and assign a discrete value which will then be used as input for our distance measure. The categories we propose are described in Table 4.5. Figure 4.3 clearly shows that, using these categories, the significance of a difference in the number of committed crimes decreases with this number. Using this table the number of crimes value is bounded by 0 and 4, while the categories maintain an intuitive treatment of both one-time offenders and career criminals. Naturally, the difference between two individuals also ranges from 0 to 4.

A large part of the database described in Appendix B contains one-time offenders.

Table 4.5: Number of crimes categories

Category	*Number of Crimes (#)*	*Assigned Value (FV)*
0	0	0
1	1	1
2	2–5	$2 + \frac{\#-2}{3}$
3	5–10	$3 + \frac{\#-5}{5}$
4	>10	4

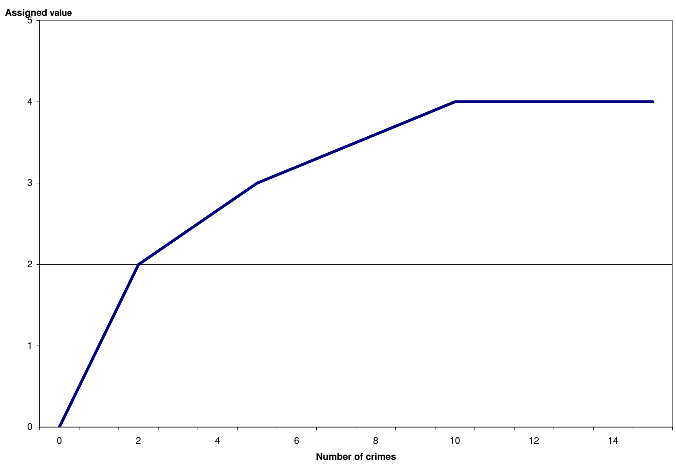


Figure 4.3: Relation between number of crimes and the assigned value (FV)

Their respective criminal careers are obviously reasonably similar, although their single crimes may differ largely in category or severity class. However, when looking into the careers of career criminals there are only minor differences to be observed in crime frequency and therefore the descriptive value of profile becomes more important. Consequently, the dependency of the *PD* on the crime frequency must become apparent in our distance measure. To this end we will multiply the profile oriented part of our distance measure with the frequency value and divide it by 4, to regain our original range between 0 and 5. Hence, the distance measure that calculates the *career difference per year* for offenders x and y ($CDPY_{xy}$) is as follows:

$$
\begin{aligned}
CDPY_{xy} &= \frac{\frac{1}{4}(PD_{xy} \cdot FVD_{xy}) + FVD_{xy}}{9} \\
&= FVD_{xy} \cdot \left(\frac{PD_{xy}}{36} + \frac{1}{9} \right)
\end{aligned}
$$

Dividing by 9, as is done above, will result in a distance between 0 and 1 for all pairs of offenders, which is standard for a distance matrix. Usage of this distance measure will result in a distance matrix that contains all distances per year for all couples of criminals. The final preparation step will consist of calculation of all these matrices for all available years and incorporating the duration of the careers. Now that we have defined a measure $CDPY^i_{xy}$ to compare two persons x and y in a particular year i, we can combine this into a measure to compare full "careers". The simplest option is to use

$$
\mathrm{dist1}(x, y) = \frac{1}{years} \sum_{i=1}^{years} CDPY^i_{xy},
$$

where *years* is the total number of years. This option is further explored in [8]. The idea can be easily generalized to

$$
\mathrm{dist2}(x, y) = \sum_{i=1}^{years} w(i)\, CDPY^i_{xy},
$$

where the weights $w(i)$ must satisfy $\sum_{i=1}^{years} w(i) = 1$. For instance, the first year might have a low weight. Of course, dist1 is retrieved when choosing $w(i) = 1/years$ for all i.

Two problems arise from this situation. The first is that careers may be the same, but "stretched". Two criminals can commit the same crimes, in the same order, but in a different time period. This may or may not be viewed as a different career. On the one hand it is only the order of the crimes that counts, on the other hand the time in between also matters. The second is that if a criminal does not commit a crime in a given year, the distance to others (for that year) will be in some sense different from that between two active criminals.

In order to address these remarks, we propose a *stretch factor* δ_S, with $\delta_S \geq 0$. This factor can be set or changed by the career analyst if he or she feels that the provided result does not fully represent reality. If $\delta_S = 0$ we retrieve the original measure, if δ_S is high many years can be contracted into a single one. To formally define our measure, we look at two careers, where for simplicity we use capitals A, B, C, $\ldots$ to indicate different crime types, and use multiset notation to specify the crimes in each year (so, e.g., $\{A, A, B\}$ refers to a year where two A-crimes and one B-crime were committed). Suppose we have the following 7 year careers x and y of two criminals:

$$
\begin{aligned}
x &= (\{A, A, B\}, \{C\}, \{A\}, \{B\}, \{D\}, \emptyset, \emptyset), \\
y &= (\{A, B\}, \emptyset, \{D\}, \emptyset, \{A\}, \{B, D\}, \emptyset).
\end{aligned}
$$

The second career may be viewed as a somewhat stretched version of the first one, The differences are: x has two As in the first year, whereas y has only one; x has the singleton set $\{C\}$ in year 2, whereas y has $\{D\}$ in year 3; and year 6 from y is spread among year 4 and 5 of x. Evidently we are looking for some sort of *edit distance*.

Now we take $\delta_S \in \{0, 1, 2, \ldots\}$ and collapse at most δ_S years, for both careers, where we add $\emptyset$s to the right. Note that, when collapsing two sets, it is important to decide whether or not to use multisets. For instance, with $\delta_S = 2$, we might change y to

$$y' = (\{A, B\}, \{D\}, \emptyset, \{A, B, D\}, \emptyset, \emptyset, \emptyset).$$

If we have t years, there are

$$t + \binom{t-1}{2} + \ldots + \binom{t-1}{\delta_S}$$

ways to change a single career; by the way, many of them give the same result. For the above example, with $t = 7$ and $\delta_S = 2$, this gives 22 possibilities, and therefore $22 \times 22 = 484$ possible combinations of the two careers. We now take the smallest distance between two of these as the final distance. For the example this would be realized through

$$\begin{aligned} x' &= (\{A, A, B\}, \{C\}, \{A\}, \{B, D\}, \emptyset, \emptyset, \emptyset), \\ y'' &= (\{A, B\}, \{D\}, \{A\}, \{B, D\}, \emptyset, \emptyset, \emptyset). \end{aligned}$$

So finally the distance of x and y is determined by the distance between the singleton sets $\{C\}$ and $\{D\}$, and the way how the multiset $\{A, A, B\}$ compares to $\{A, B\}$. This last issue can be easily solved in the basic computation of *PD*, discussed in Section 4.4.

Of course, there are other practical considerations. For instance, some (perhaps even many) crimes will go unnoticed by police forces, giving an incorrect view. Furthermore, in the current approach there is a clear, but maybe unwanted separation between crimes committed in December and in January of the next year.

4.6 Visualization

The standard construction of our distance matrix enabled our tool to employ existing techniques to create a two-dimensional image. The intermediate Euclidean distances in this image correspond as good as possible to the distances in the original matrix. Earlier research [11] showed that automated clustering methods with human intervention might get promising results. The method that we incorporated in our tool was described in Chapter 4 and [5], where we built upon the push-and-pull architecture described in [20], and allows data analysts to correct small mistakes made by naive clustering algorithms that result in local optima. The user is assisted by coloring of the nodes that represent how much a single individual or a collection of individuals contribute to the total error made in the current clustering. The end result is a clickable two-dimensional image of all possible careers. When the user clicks on an individual he or she is presented with the profile information for that particular offender.

A standard K-means algorithm was used to automatically find clusters in the visualization. These clusters correspond to the large groups in the visualization and can be seen as an internal model of the visualization. If such a cluster is viewed as a representation of a certain class of criminal careers the internal clustering can be used to classify new individuals in the visualization by finding out in which cluster they reside. Note that an incremental visualization method is needed to correctly add such a new individual in the first place.

4.7 Prediction

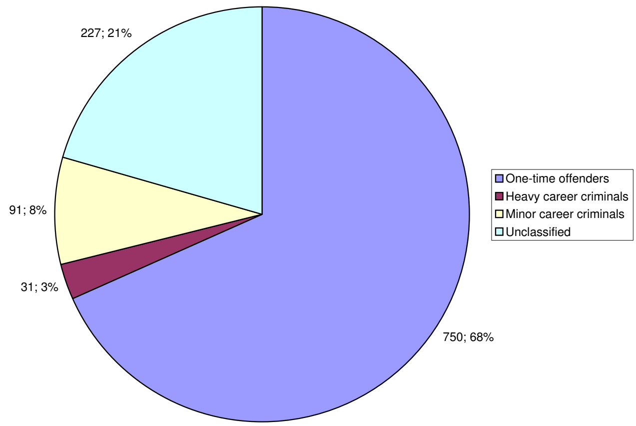
Once a (large) group of criminals has been analyzed, it is possible to predict future behavior of new individuals. To this end we can readily use the tool from Chapter 3, [5]. Suppose that the given database has already been converted into a two-dimensional visualization and a clustering diagram has been created, as described in Section 4.6. We can and will assume that the original database consists of "full" careers. For a new partially filled career, e.g., consisting of only two consecutive years with crimes, we can compute its distance to all original ones taking into account only the number of years of the new career. The system now takes care of finding the right spot for this individual, where it reaches a local minimum for the distances in the two-dimensional space in comparison with the computed distances. The neighboring individuals will then provide some information on the possible future behavior of the newly added career; this information is of a statistical nature.

Of course, in practice it is not always clear when a career ends, not to speak of situations where a career ends or seems to end prematurely, e.g., because a criminal dies or is in prison for a substantial time period. More on the possibilities concerning prediction can be found in Chapter 5, where some initial steps are taken to facilitate extrapolation and in Chapter 7 and Chapter 8, where an actual attempt is made at the prediction of criminal careers based upon this data.

4.8 Experimental Results

We tested our tool on the actual database described in Appendix B, containing approximately one million offenders and the crimes they committed. Our tool analyzed the entire set of criminals and presented the user with a clear two-dimensional visualization of criminal careers as can be seen in Figure 4.5.

Three different clusters were automatically identified. The largest of them, seen in the bottom of Figure 4.5, was easily identified as the group of one-time offenders. The other two visible clusters were identified through manual sampling, investigating the four factors of the criminal careers of the sampled perpetrators in the cluster under observation. Combined with a default state of "unclassified", we can distinguish four different classes of criminal careers, proportioned as described in Figure 4.4, where the amounts are specified in thousands.



Career Class	*Percentage*
One-time	68%
Heavy career	3%
Minor career	8%
Unclassified	21%

Figure 4.4: Proportion of offenders in each class of criminal career

The confidential nature of the data used for criminal career analysis prevents us from disclosing more detailed experimental results reached in our research.

The image in Figure 4.5 gives an impression of the output produced by our tool when analyzing the before mentioned database. This image shows what identification could easily be coupled to the appearing clusters after examination of its members. It appears to be describing reality very well. The large “cloud” in the left-middle of the image contains (most of the) one-time offenders. This seems to relate to the database very well since approximately 75 % of the people it contains has only one felony or misdemeanour on his or her record. The other apparent clusters also represent clear subsets of offenders. There is however a reasonably large group of “un-clustered individuals”. The grouping of these individual criminal careers might be influenced by the large group of one-timers. Getting more insights into the possible existence of subgroups in this non-cluster may prove even more interesting than the results currently provided by our approach. Future research will focus on getting this improvement realized (cf. Section 4.9).

4.9 Conclusion and Future Directions

In this chapter we demonstrated the applicability of data mining in the field of criminal career analysis. The tool we described compiled a criminal profile out of the four

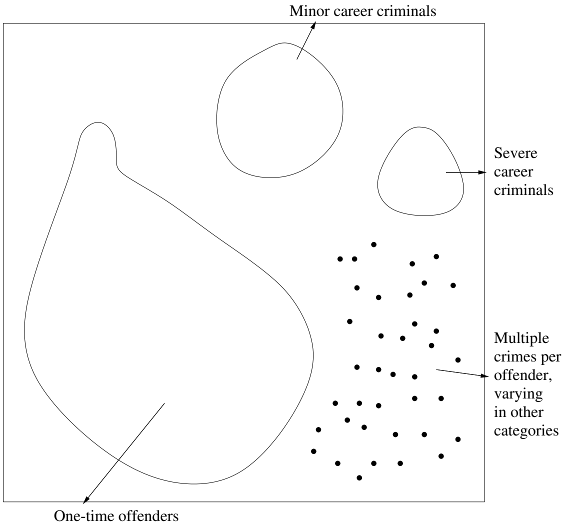


Figure 4.5: Visualization of Criminal Career clusters

important factors describing a criminal career for each individual offender: frequency, seriousness, duration and nature. These profiles were compared on similarity for all possible pairs of criminals using a new comparison method. We developed a specific distance measure to combine this profile difference with crime frequency and the change of criminal behavior over time to create a distance matrix that describes the amount of variation in criminal careers between all couples of perpetrators. This distance measure incorporates intuitive peculiarities about criminal careers. The end report consists of a two-dimensional visualization of the results and is ready to be used by police experts.

The enormous “cloud” of one-time offenders gave a somewhat unclear situational sketch of our distance space. This problem, however, can not be easily addressed since a large part of the database simply consists of this type of offenders. Its existence shows, however, that our approach easily creates an identifiable cluster of this special type of criminal career, which is promising. One possible solution to this problem would be to simply not take these individuals into account when compiling our distance matrices.

The used method of visualization and clustering provided results that seem to represent reality well, and are clearly usable by police analysts, especially when the above is taken into account. However, the runtime of the chosen approach was not optimal yet. The visualization method was too intensive in a computational way, causing delays in the performance of the tool. In the future, an approach like Progressive Multi-Dimensional Scaling [35] could be more suited to the proposed task in a computative way, while maintaining the essence of career analysis.

Future research will aim at solving both concerns mentioned above. After these issues have been properly addressed, research will mainly focus on the automatic comparison between the results provided by our tool and the results social studies reached on the same subject, in the hope that “the best of both worlds” will reach even better analyzing possibilities for the police experts in the field. Incorporation of this tool in a data mining framework for automatic police analysis of their data sources is also a future topic of interest.

An inherit issue with these kind of approaches is that applicability in courts and daily police practice depends on statistical insightfulness and relevance, current law and that the fact that they are subject to privacy considerations. These matters are discussed in detail in Appendix A.

Chapter 5

Enhancing the Automated Analysis of Criminal Careers

Four enhancements have been devised, both on a semantic and efficiency level, that improve existing methods of automated criminal career analysis described in Chapter 4. A new distance measure was introduced that more closely resembles the reality of policing. Instead of the previously suggested, more rigid, comparison of career changes over time we propose an alignment of these careers. We employ a faster and better method to cluster them into a two-dimensional plane. This visualization can ultimately be used to predict the unfolding of a starting career with high confidence, by means of a new visual extrapolation technique. This chapter discusses the applicability of these new methods in the field and shows some preliminary results.

5.1 Introduction

In Chapter 4 (cf. [7]), we described research attempting to gain insights into the concept of criminal careers: the criminal activities that a single individual exhibits throughout his or her life. The resulting tool analyzed the criminal record database in described in Appendix B. This method mainly addressed the extraction of the four important factors (see Section 5.2) in criminal careers and established an overview picture on the different existing types of criminal careers by using a stepwise or year-based approach, with the ultimate objective to prepare the data for prediction of a new offender's career. The approach centered on formalizing an individual's *criminal profile* per year, representing an entire career as a string of these calculated profiles. Chapter 4 identified some difficulties in comparing these strings and provided solutions to them. The method, however, suffered from time-complexity issues and a comparison mechanism that was largely rigid and ad hoc.

In this chapter we describe a number of techniques were specifically developed for the enhancement of the above mentioned analysis but are also widely applicable in other areas. We explain how these methods can be used to improve the existing paradigm by replacing the individual ad hoc methodologies and how combining them will reduce the number of steps needed to reach better results. We supplement the existing approach by adding a preliminary prediction engine that employs the properties of the already realized visualization. Experiments performed with this setup are also discussed.

The main contribution of this research lies in the novel combination of a number of separately created technologies, all very well suited for law enforcement purposes, to pursue the common goal of early warning systems that allow police agencies to prevent the unfolding of possibly dangerous criminal careers.

Four enhancements have been devised. In Section 5.3 a new distance measure is introduced that more closely resembles the reality of policing. Instead of previous, more rigid, comparison of career changes over time we propose an alignment of these careers in Section 5.4. In Section 5.5 we employ a faster and better method to cluster them into a two-dimensional plane. This visualization can ultimately be used to predict the unfolding of a starting career with high confidence, by means of a new visual extrapolation technique (see Section 5.6). Section 5.7 shows experiments, and Section 5.8 concludes.

5.2 Background and Overview

In Chapter 4 the four factors were extracted from the criminal record database, recorded as numbers and treated as such. However, it appears to be beneficial to treat a collection of crimes in a single year as a *multiset*, which then describes severity, nature and frequency inherently. A multiset or *bag*, is a collection where each element can occur more than once. The set of all distinct elements in that multiset is called its *underlying set*. Figure 5.1 describes the relation between the two and shows how we employ it to represent a criminal's activities in a single year.

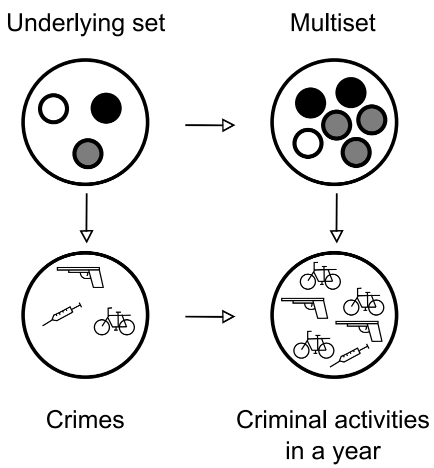


Figure 5.1: A multiset representation of a criminal profile in a single year

The multiset representation offers advantages, most notably the availability of standard approaches to compare multisets and calculate distances between them. Kosters and Laros [18] devised a distance function for multisets that generalizes well-known distance measures like the *Jaccard distance* [17]. This metric contains a customizable function f that can be adapted to fit specific knowledge domains. It also allows for the incorporation of weights for an element, e.g., element x counts twice as much as element y. We employ this metric for calculating the difference between two crime-multisets (see Section 5.3) by choosing a specific f.

Instead of the former method of a strict number-wise comparison between years (comparing the first year of criminal a with the first year of criminal b, the second year of a with the second year of b, etc.), with the possibility of stretching or shrinking careers (cf. Section 4.5, we propose a novel *alignment* of the mentioned multisets. This method strives for an optimal automated matching of years, using the distance measure described above, assigning penalties for every mutation needed, which enables a police analyst to better cope with situations like captivity, forced inactivity or unpenalized behavior. Section 5.4 elaborates on this.

Visualization and clustering methods used before yielded good results, mainly by incorporating direct input from police specialist. It was however computationally complex, taking a very long time to process the entire dataset. Within the scope of the research into criminal careers we developed a method that improved standard *push-and-pull* algorithms but eliminated the need for human input, while retaining the former strength of its output [19]. We explain this in Section 5.5.

Earlier results provided some important information for law enforcement personnel, but the ultimate goal, predicting if certain offenders are likely to become (heavy) career criminals, was not realized or elaborated upon. Further investigation of this possibility led to the need of a good two-dimensional visual *extrapolation* system, described in [12]. The conceivement of this technique paved the way for possible development of early warning systems, that we explore in Section 5.6.

5.3 A Distance Measure for Profiling

The core of our approach consists of a new distance measure. This metric is a special case of the *generic metric for multisets* as described in [18]:

$$d_f(X, Y) = \frac{\sum_i f(x_i, y_i)}{|S(X) \cup S(Y)|}$$

This metric calculates the distance between two finite multisets X and Y, where $S(A) \subseteq \{1, 2, \ldots, n\}$ is the underlying set of multiset A and a_i is the number of occurrences of element i in multiset A. Here $f : \mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is a fixed function with finite supremum M and the following properties:

$$\begin{array}{lcll} f(x,y) & = & f(y,x) & \text{for all } x,y \in \mathbb{R}_{\geq 0} \\ f(x,x) & = & 0 & \text{for all } x \in \mathbb{R}_{\geq 0} \\ f(x,0) & \geq & M/2 & \text{for all } x \in \mathbb{R}_{>0} \\ f(x,y) & \leq & f(x,z) + f(z,y) & \text{for all } x,y,z \in \mathbb{R}_{\geq 0} \end{array}$$

These properties ensure that d_f is a valid metric [18].

Naturally, all results depend largely upon choosing the right *defining function* f for a certain knowledge domain. In the law enforcement area, it is important to realize that the relative difference between occurrences of a crime is more important than the difference alone. This is because the distance between an innocent person and a one-time offender should be much larger than for example the distance between two career criminals committing 9 and 10 crimes of the same kind respectively, thus ensuring that $f(0, 1) \gg f(9, 10)$ (the two arguments of f are the numbers of respective crimes of a given category, for two persons).

A good candidate for this function, that was developed in cooperation with police data analysts, seems to be the function

$$f_{crime}(x, y) = \frac{|x - y|}{(x + 1)(y + 1)}$$

for integer arguments x and y, both ≥ 0. This function complies with the above mentioned characteristic of criminals and yields a valid metric.

It is obviously important to still be able to incorporate *crime severity* and *nature*, next to crime frequency, into this equation. This is possible through the addition of *weights* to the generic metric described above. A suggestion was made by Kosters and Laros [18] for accomplishing this: each item i gets an assigned integer weight ≥ 1 and is multiplied by this weight in every multiset, simulating the situation where all weights were equal but there were simply more of these items in each set. An impression of how this affects the calculation of distances between criminal activities is given in Figure 5.2.

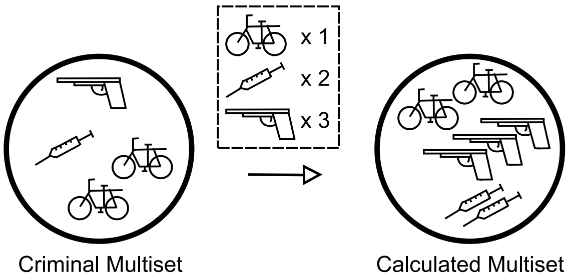


Figure 5.2: Adding weights to criminal activities

By using this specifically tailored distance measure with weighing possibilities, we are able to calculate distances between criminals per single time frame. This distance can serve as basis to discover distances between careers, that can be seen as strings of these time frames.

5.4 Alignment of Criminal Careers

Since the temporal ordering of crimes is of high importance in the definition of a criminal career, a tool occupied with its analysis should provide a way to calculate distances between these ordered series based upon distances between their elements. Sequence alignment could be a valuable tool in this process.

In contrast with the strict year-by-year comparison used in Chapter 4, the alignment paradigm [21] tries to find a best match between two ordered series, allowing a small number of *edits* to be made in such an ordering: insertions, deletions and other simple manipulations. The penalty for a deletion or insertion is governed by a so-called *gap-penalty function*. Alignment is an algorithm typically used within the area of computational biology. Famous instances include those of Needleman-Wunsch [23] and Smith-Waterman [27]. Each alignment is based upon a valid metric for elements (a year of accumulated crimes in this case), as for example the metric discussed in Section 5.3. Figure 5.3 describes a typical alignment situation, showing two such edits. The treatment of gaps, however, may somewhat differ from biological applications; also note that empty multisets (an "innocent" year) have a special relation with the gap penalty. Next to these differences, careers may or may not be "complete"; some of them are still unfolding.

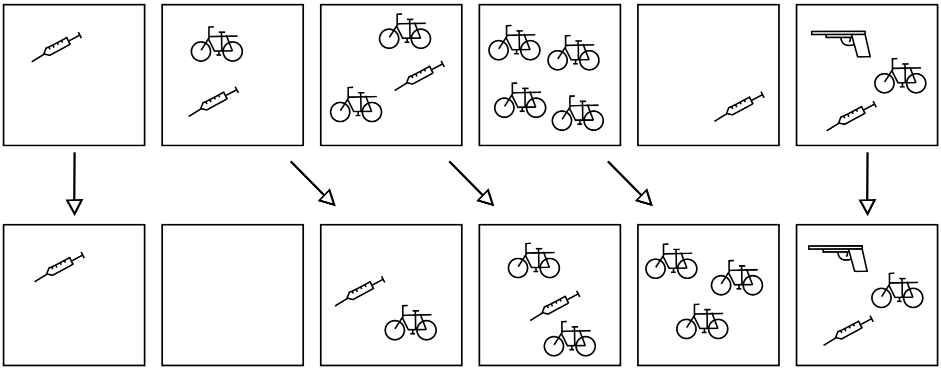


Figure 5.3: Two criminal careers who's similarity is revealed by alignment

One of the rationales for using alignment instead of strict sequence matching is the frequent occurrence of gaps or stretches in criminal data. One could think of, for example, prison time, forced inactivity, judicial struggles consuming time between an offense and its penalization and, most importantly, the frequent occurrence of unpenalized criminal activity due to undiscovered crime. Treating this naturally occurring plethora of phenomena without any descriptive value on one's intentions in the criminal area as non-existent or not important will result in a deformed image of compared criminal careers. Therefore the usage of an alignment mechanism is strongly favored over a strict comparison on a semantic level. Another reason for this might be the occurrence of randomly appearing year changes within one's career, e.g., two crimes occurring either in June and July, or in January and December, are in essence the same, although strict separation between years will place the latter in two different time frames. Although data is often gathered in such broad time frames, leading to the inevitable occurrence of this problem, the alignment paradigm could potentially be used to reduce the subsequent effects.

Careful consideration needs to be given to the gap-penalty function in the field of crime analysis. While each gap in standard (biological) alignment procedures represents a constant penalty, a gap in a criminal career can be considered less or more important based upon the amount of time passed within this gap. We therefore assign a time stamp $t(a)$ to each element a in the criminal career. The gap-penalty is then computed by applying the gap-penalty function to the different $\Delta t(u, i) = t(u_{i+1}) - t(u_i)$ involved, where u_i is the i^{th} element in u, an ordered, time stamped series. The known algorithms to compute distances have to be adapted, causing an increase in time ($O(n^2) \rightarrow O(n^3)$) and space complexity ($O(n) \rightarrow O(n^2)$), where n is the length of the series.

Using this alignment we compare all possible couples of criminal careers and construct a standard distance matrix of the results.

5.5 Toward a New Visualization Method

In the original setup, a *push-and-pull algorithm* [5] was used to create a (sub)optimal two-dimensional visualization of the produced distances matrix from a randomized starting point. Initially, points get random positions in the unit square of the plane; after which they are repeatedly pulled together or pushed apart, depending on the relation between current and desired distance. This method, a variant to Multi-Dimensional Scaling, relied upon domain knowledge provided by the analyst using the tool. This domain input realized a significant gain in results compared with simpler, unsupervised algorithms of the same kind [20].

The complexity of the previous algorithm, however, prohibited a time efficient usage of the tool, costing near days on standard equipment to analyze the target database and thus posing a serious bottleneck in the original setup that needed to be overcome, preserving the property that elements can be easily added and traced.

One of the major problems of the simpler variants of the push-and-pull algorithm was the fact that a large number of randomly positioned individuals tended to create subgroups that pushed a certain item harder away from its desired position in the plane, than comparable items pulled this point toward that position (see Figure 5.4, standard).

Especially in the case of criminal careers, where analysis tends to result in vastly different, very large clusters, this effect appears to present. Addressing this issue could therefore lead to a representative visualization and good response times, eliminating the need for human input. Kosters and Laros [19] suggested the usage of a *torus* rather than the standard bounded flat surface. Within such a torus, all boundaries are identified with their respective opposite boundaries, enabling the "movement" of an individual from one end of the surface to the opposite end, thus overcoming the above mentioned problem (Figure 5.4, torus).

When using this torus to construct a two-dimensional visualization of our distance matrix, time complexity was reduced to the level of simple push-and-pull algorithms, but the visualization is expected to be of the same quality as the image produced by the supervised algorithm. Other MDS techniques are also potential candidates for adoption, if they adhere to the demand of incremental addition of single items.

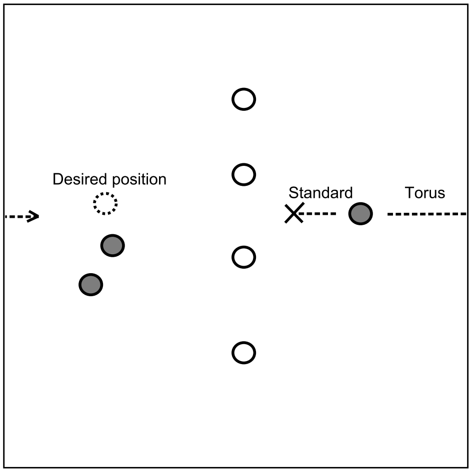


Figure 5.4: Advantage of torus visualization versus standard visualization

5.6 Prediction of Unfolding Careers

As was mentioned in Chapter 4, a lot of information is inherently present in the visualization of mutual distances. In Chapter 7, [12] we propose to utilize the power of a two-dimensional visualization for temporal extrapolation purposes. This algorithm plots the first, already known, time frames of criminal activity of a certain person in the correct place (compared to other fully grown careers already displayed in the image) and extrapolates these points in the plane to automatically discover careers that are likely to become very similar in the future. A system employing this should be able to provide police operatives with a warning when such a starting career can easily develop into that of a heavy career criminal. More on the possibilities concerning prediction can be found in Chapters 7 and 8, where an actual attempt is made at the prediction of criminal careers based upon this data.

5.7 Experimental Results

We tested our new approach on the criminal record database (cf. Appendix B), containing approximately one million offenders and the crimes they committed. Our tool analyzed the entire set of criminals and presented the user with a clear two-dimensional visualization of criminal careers as can be seen in Figure 5.5.During this process, a multiset string of offenses was added to each offender, where each bucket contained crimes committed within a certain time frame.

The image in Figure 5.5 gives an impression of the center of the torus output produced by our tool when analyzing the before mentioned database. This image shows the identification that could easily be coupled to the appearing clusters after examination of

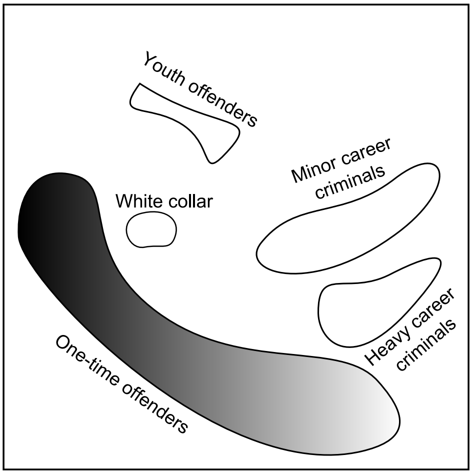


Figure 5.5: Impression of clustered groups of criminal careers in the visualization

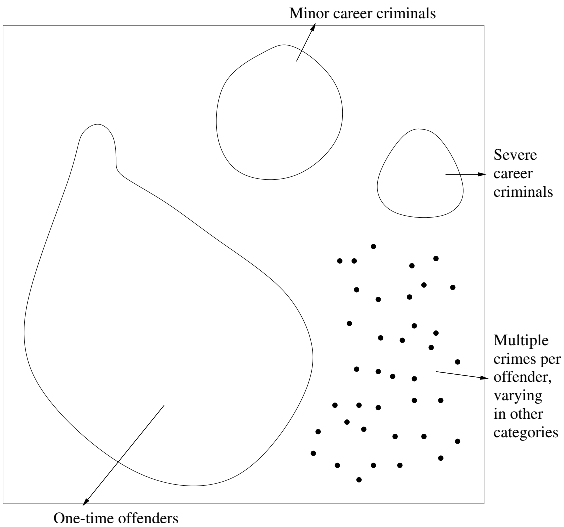
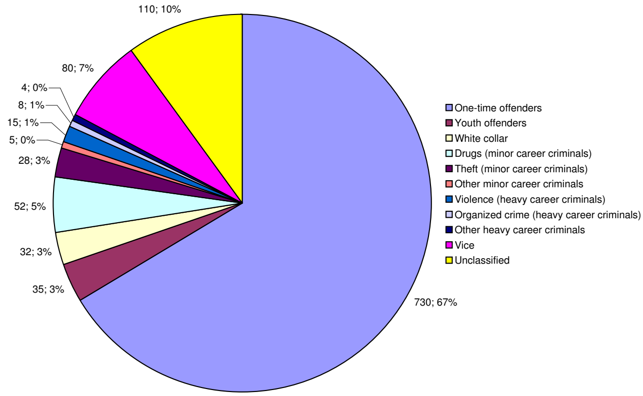


Figure 5.6: Results from Chapter 4

its members. In total 10 different clusters were identified using a standard k-means clustering method, of which some were left out of Figure 5.5, because they were to small or because a combination of clusters provided a visualization that was easier to interpret. Below, in Figure 5.7 they are specified in more detail, accompanied by the distribution of all offenders from the database over these classes, where amounts are specified in thousands.

These figures appear to be describing reality better than the visualization resulting



Career Class	*Percentage*
One-time	66%
Youth Offenders	3%
White Collar	3%
Drugs (Minor Career)	5%
Theft (Minor Career)	3%
Other Minor Career	1%
Violence (Heavy Career)	1%
Organized Crime (Heavy Career)	1%
Other Heavy Career	1%
Vice	7%
Unclassified	10%

Figure 5.7: Proportion of offenders in each class of criminal career

from Chapter 4, [7]. Just like the image constructed by the previous approach (Figure 5.6) the new image features a large "cloud" of one-time offenders. However a clear distinction can now be noticed between the left, dark, part of that cloud, which represent minor crimes, and the right, lighter side of the cluster that contains the heavier felonies. Next to this, the group of miscellaneous offenders was split into two, more specific, groups; the white border criminals and youth offenders. The remaining unclustered individuals were left out of the image for clarity reasons. Analysis of this group can however also reveal interesting results, answering why these individuals do not belong to one of the larger clusters and what kind of crimes these people commit. Next to the better results provided by this approach, far better computation times were realized that outperform the previous method by a factor 100.

Getting more insights into the possible existence of subgroups in any cluster remains a desirable functionality of this approach. Future research will focus on getting this improvement realized (cf. Section 5.8). Next to that, it might be of interest to find common subcareers in the complete list of careers, to find out if there are any subcareers that occur often. An even more important search would be to find careers that are common in one class of offenders and are not in others, in a way representing the class they are common in. These possibilities are discussed in Chapter 6.

5.8 Conclusion and Future Directions

In this chapter we demonstrated the applicability of some newly developed methods in the field of automated criminal career analysis. Each of the enhancements provided a significant gain in computational complexity or improved the analysis on a semantic level. An integral part of the new setup consists of the multiset metric that was specifically tuned toward the comparison of criminal activities. It leaned strongly upon knowledge provided by domain experts. This distance measure was used to search for optimal alignments between criminal careers. This edit distance for sorted or temporal multiset data improved greatly upon the original year-by-year comparison, dealing with problematic situations like captivity or forced inactivity. Using this alignment a distance matrix can be constructed and clustered on a flat surface using a new unsupervised method, yielding comparable results with the previous setup, but largely reducing the time complexity. The power of this visual representation was employed to extrapolate a starting career within the plane of the visualization, predicting its unfolding with some accuracy. The end report consists of a visual two-dimensional visualization of the results and a prototype of an early warning system that predicts newly developing criminal careers, which is ready to be used by police experts.

This chapter describes improvements on a methodology for the analysis of criminal careers introduced in Chapter 4. Building upon a series of successfully introduced algorithms, this novel approach is subject to a high number of parameters and leans heavily on the way these techniques work together. It is to be expected that the current setup can also be improved by tweaking its parameters and elaborating on the internal cooperation between phases. Future work will therefore focus on a larger experimental setup to find and verify optimal settings and retrieve results that are even more descriptive and usable. We also hope to equip our tool with a sub-cluster detection algorithm to provide even better insights into the comparability of criminal careers.

It may be of interest to set fuzzy borders between the different years. Crimes within months ending or beginning such a time frame can be (partly) assigned to the next or previous year respectively as well, thus eliminating the problems arising with strict coherence to the change of calendar year.

Special attention will be directed toward the predictability of criminal careers, and the eventual suitability of this approach for early warning systems running at police headquarters throughout the districts. Incorporation of this new tool in a data mining framework for automatic police analysis of their data sources is also a future topic of interest.

As this chapter describes an extension to the material described in Chapter 4, the same privacy and judicial constraints apply to the matter discussed here. They are discussed in detail in Appendix A.

Chapter 6

Detection of Common and Defining Subcareers

The search for common sequences in a database of timed events is a relatively recent addition to the data mining research area. It usually aims to provide insights into customer behavior, based upon the sequential acquisition of certain products, but is also applicable to criminal careers. A method for the automatic categorization of criminal careers has been established in Chapter 5 and using this classification, a method was devised that can find subcareers that can be said to be specific for one class, based upon its occurrence rate in this class and the other classes. It builds upon a well-known approach to search for common subsequences in databases. In this chapter we propose a number of key-changes to this approach that make it applicable to the search of common criminal careers. We propose to expand this method with the search of defining subcareers and show some experiments that investigate the possibilities of sequence mining in this area of law enforcement.

6.1 Introduction

Now that a method has been established for the classification of criminal careers and a solid basis has been created upon which prediction can be performed, it might also be of interest to discover if there are any chains of criminal activity that occur often. Such a search for common subcareers might both be of interest to the field of crime analysis at criminology institutes or police headquarters, to a police officer, who can use the knowledge of commonly occurring patterns in crime to his advantage in the field and to social workers at judicial institutions, dealing with starting offenders, who can employ knowledge of commonly occurring patterns in their treatment of delinquents.

Besides the mere discovery of these possibly occurring common subcareers, it is of importance to the causes mentioned above to establish a relationship between some of these common subcareers and their respective existence in different criminal career classes as described in Chapter 4 and Chapter 5. If such distinguishing or defining subcareers can be found with a certain threshold, a reliability can be established that couples the occurrence of these subcareers to a certain class, enabling law enforcement personnel to quickly recognize and impede these behavioral patterns.

For this purpose we are given the criminal record database described in Appendix B, containing a list of crimes per offender organized per time frame (year or month). Obviously the number of occurrences of each single crime type within one time frame is important in this matter, especially when the time frames are rather large.

We introduce the problem of mining sequential patterns over this data. An example of such a pattern is that offenders typically first come in contact with law enforcement for possession of drugs, then minor theft, then getting involved in drug dealing felonies. Naturally, elements of such a sequence need not be single crimes, they can be sets of crimes or even sets of the same crime. Furthermore, the patterns should be consecutive, meaning that for an offender to satisfy the inclusion demands for a pattern, each of the included crimes must be committed right after one another or within the same time frame. Compliance with this demand guarantees that a sequence “Bicycle theft”→“Drug abuse” is treated differently than “Bicycle theft”→“Murder”→“Drug abuse”, which is correct in the area of crime analysis, since both criminal careers are treated very differently.

Since sequential pattern mining has been researched for a while, a background to our work is provided in Section 6.2. The main contribution of this chapter is in Section 6.3 where the alterations to standard approaches are discussed that lead to successful subcareer mining. Section 6.4 shows some of the results reached by our approach when applied on the criminal record database.

6.2 Background

A lot of work has already been done in the area of sequential pattern mining, commonly motivated by decision support problems from the retail world. The authorative work on this subject is [2], where sequential consumer behavior was first automatically analyzed. This paper dealt with the analysis of sequentially bought *baskets*, sets of items bought in a single batch, and was a continuation of the work in [1], where the focus was more on intra-basket patterns, and the discovery of what items are commonly often bought together. The work contained a mixture of intelligent systems that predicted common continuation of a series [9, 36], instead of finding all common patterns, of systems that attempt to find text subsequences based upon regular expressions [3, 25, 34, 32] and on the discovery of similarities in a database of genetic sequences [33].

Although this work is archetypical for the field, its algorithms are still widely applicable and are very well suited as a basis for the problem under observation. There are however a number of critical issues that need to be dealt with:

1. The problem in [2] deals with itemsets, where each item is either present or not, while in the problem at hand, each crime can be present multiple times, therefore dealing with the multiset paradigm.

2. In the original problem, the items in the target pattern were not required to appear consecutively in a sequence. In contrast however, as stated above, within criminal career analysis, the items *must* be consecutive to fulfill the requirement. Within our approach we ignore possible empty time frames (gaps).

3. The "boundaries" between time frames have no implicit meaning for subcareer detection other than the fact that they separate groups of crimes of which the ordering is not known, which arises from the fact that crimes are only added to the database periodically, losing their original ordering within these periods or time frames. Consequently, the discovery of subcareers takes place on sequences of single crimes, where parts of these sequences can be ordered in all possible ways. This sharply contrasts with the original method where boundaries between the different itemsets were not to be broken during the pattern matching process.

These issues will be addressed in Section 6.3.

Usually a multiset of crimes is denoted by $\{i_1, i_2, \ldots, i_m\}$ where i_k is a crime ($1 \leq k \leq m$) such that elements can occur more than once; we will denote this multiset by $(i_1\ i_2\ \ldots\ i_m)$. A criminal career is then denoted by $\langle s_1\ s_2\ \ldots\ s_n \rangle$, where s_k is a multiset of crimes representing a single time frame ($1 \leq k \leq n$). According to [2] a sequence $\langle a_1 a_2 \ldots a_n \rangle$ is *contained in* another sequence $\langle b_1 b_2 \ldots b_m \rangle$ if integers $1 \leq i_1 < i_2 < \ldots < i_n \leq m$ exist such that $a_1 \subseteq b_{i_1}, a_2 \subseteq b_{i_2}, \ldots, a_n \subseteq b_{i_n}$. This means, for example, that $\langle (1)\ (2\ 3)\ (4) \rangle$ is contained in $\langle (0)\ (1\ 11\ 12)\ (2\ 3\ 10\ 99)\ (55)\ (4) \rangle$, but that $\langle (1)\ (2) \rangle$ is not contained in $\langle (1\ 2) \rangle$. Clearly, this last effect conflicts with our third requirement, mentioned above, hence we have to resort to another construct within our approach. A sequence is said to be *maximal* in a certain set of sequences, if it is not contained in another sequence from that set.

Using the definition above, a sequence from the database *supports* a sequence if it contains that sequence. The *support* of a certain (sub)sequence can then be defined as the fraction of total database sequences that support this sequence. Mining for common subsequences can thus be seen as the search for maximal subsequences that have a certain threshold support as defined by the analyst. These subsequences are called *frequent*.

As an example, we examine the data in Table 6.1, where a crime type is denoted as a number. If these were the only active periods of criminal activity of these offenders, their criminal careers would be described by Table 6.2, where the middle column contains the notation as laid out in [2]. In the third column we introduce a notation that is more suited to our approach, where the boundaries between time frames have been discarded and where a multiset has been overlined, meaning that its elements can be randomly ordered. Note that for clarity singletons are not overlined. The new notation is equivalent with the original notation of customer transactions Henceforward, we will use the new notation to *denote* that these series need to comply with the three requirements specified above.

Assuming a support threshold of 25%, meaning a minimum support of two offenders, the sequences $\langle (1)\ (8) \rangle$, $\langle (2)\ (4) \rangle$ and $\langle (5)\ (6\ 7) \rangle$ are the maximal frequent subsequences using the middle column. Note that $\langle (5)\ (6) \rangle$ and $\langle (5)\ (7) \rangle$ also satisfy the threshold, but since they are not maximal they are not considered as end-result.

If we consider the situation denoted in the third column, 18 is no longer frequent,

Table 6.1: Database of criminal activity sorted by Offender and Time frame

Offender	*Time frame*	*Crimes*
1	September	1 2
1	October	4
1	December	8
2	October	2 6 9
3	October	1 4 5
3	November	6 7 8
3	December	1 1
4	November	5
4	December	6 7
5	September	2 3
5	November	3 4

Table 6.2: Criminal career version of Table 6.1

Offender	*Career (cf.* [2]*)*	*Career (our approach)*
1	$\langle(1\ 2)\ (4)\ (8)\rangle$	$\overline{12}48$
2	$\langle(2\ 6\ 9)\rangle$	$\overline{269}$
3	$\langle(1\ 4\ 5)\ (6\ 7\ 8)\ (1\ 1)\rangle$	$\overline{145}\,\overline{678}\,\overline{11}$
4	$\langle(5)\ (6\ 7)\rangle$	$5\overline{67}$
5	$\langle(2\ 3)\ (3\ 4)\rangle$	$\overline{23}\,\overline{34}$

since crimes 1 and 8 are separated by crime 4 in the career of offender 1, however, 24 is now frequent representing both $\langle(2)\ (4)\rangle$ and the "stronger" $\langle(2\ 4)\rangle$. Finally, using the same argument, $\langle(5)\ (6\ 7)\rangle$ or $5\overline{67}$, now also contains 567. However, the overlining of $\overline{67}$ can not be removed, since this would effectively omit the containment of 576, per the consecutiveness demand. Evidently, there are quite a few significant differences between the original, retail related, approach and the search for common subcareers.

In the new approach, containment is now defined as follows: a sequence $a = a_1a_2\ldots a_n$ is contained in sequence $b = b_1b_2\ldots b_m$ if there is a subsequence $b_{i+1}b_{i+2}\ldots b_{i+n}$ with $a_1 \subseteq b_{i+1}$, $a_n \subseteq b_{i+n}$ and $a_j = b_{i+j}$ for $1 < j < n$, where $i \geq 0$, $i + n \leq m$ and a_k and b_k denote time frames. Note that the ordering in a time frame is undefined, e.g., $\overline{1123} = \overline{3121}$, since it is a multiset.

If we consider the problem of strict borders between time frames, as described in Chapter 4 and Chapter 5, where criminal activity at the end of one time frame and at the beginning of the next time frame are strictly separated even though the time between these crimes is very short, another definition of containment might be better. When investigating a subsequence in a certain time frame, we could instead consider its appearance in the merger of this time frame and both its temporal neighbors, assuming that the borders between those time frames are somewhat fuzzy. This would result in the following definition of containment: a sequence $a = a_1 a_2 \dots a_n$ is contained in sequence $b = b_1 b_2 \dots b_m$ if there is a subsequence $b_{i+1} b_{i+2} \dots b_{i+n}$ with $a_j \subseteq b_{i+j-1} \cup b_{i+j} \cup b_{i+j+1}$ $(j = 1, 2, \dots, n)$ and $b_{i+j} \subseteq a_{j-1} \cup a_j \cup a_{j+1}$, $(j = 2, 3, \dots, n-1)$, $i \geq 0$, $i + n \leq m$ and a_k and b_k are time frames and $b_{-1} = b_{m+1} = \emptyset$.

There are however two related problems when dealing with the fuzzy border situation:

1. **Empty set problem** In contrast with the first definition of containment, the existence of empty sets, which in this case represent empty time frames (a time frame where a certain individual commits no crimes), poses a problem in this case. In the first situation no consecutiveness is violated when omitting periods without activity from a career. However, if we consider borders between periods to be fuzzy, and the empty time frames are omitted from a career, we can invalidly have fuzzy borders between two time frames that were no neighbors in the original sequence. For example if we consider the career $1x2$, where x is a period without activity, there is clearly a strict separation between activity 1 and 2 (at least a time frame). However, if we denote this career as 12 and consider the border between the two to be fuzzy, a sequence 21 would invalidly be contained.

2. **Overlapping neighbors problem** A related problem is the case where an overlap between the predecessor and successor time frames of the time frame under consideration occurs, even though they are strictly separated by that time frame. For example, in the career $\overline{12}3\overline{45}$, the subcareer 52 would invalidly be contained. Even though the intention is to let $\overline{123}$ and $\overline{345}$ be considered, the combination thereof ($\overline{12345}$) should not.

These issues can be dealt with at the cost of a much more complicated containment relation, therefore, we will choose the first definition of containment within the scope of this paper.

The standard approach for the detection of frequent subsequences consists of five steps:

1. **Sort Phase.** The database is sorted using offender as major key and time frame as minor key. The database is now implicitly a database of criminal careers.

2. **Large Itemset Phase.** The set of single itemsets that are frequent (large itemsets) is retrieved. Standard frequent itemset methods are suitable for this approach, except for the fact that they use a slightly different notion of support. In these algorithms, the support of an itemset is defined as the number of transactions (or in our case

time frames) a certain itemset appears in, while in this case support should be seen of the appearances of an itemset in *any* of the transactions of a single individual. This deficit can be overcome easily if we only count the occurrence of an itemset once for each individual, even if it appears more.

3. **Transformation Phase.** The database will be transformed in such a way that the sequences or careers are replaced by sequences of frequent itemsets, discovered in the previous phase. Individuals without any frequent itemsets in their sequence are discarded, however they still contribute to the total amount of individuals. This phase is necessary to reduce the computation time when searching for frequent sequences.

4. **Sequence Phase.** The actual frequent sequences are found in this phase. Just like the search of the sequence building blocks, the large itemsets, the search for large sequences is based upon the APRIORI-property that a certain sequence can only be frequent if all its subsequences are also frequent. Therefore, this phase is a multi-pass algorithm that generates candidate large sequences based upon sequences that are already large. These candidates are then compared to the transformed database from Step 3 and only the large sequences are kept. The cycle then starts over, with the generation of candidate large sequences.

5. **Maximal Phase.** All sequences that are frequent but not maximal are discarded. This can be accomplished easily by deleting all sequences that are contained in another sequence from the set discovered in Step 4. All sequences of length 1 are also left out.

This process is visualized in Figure 6.1.

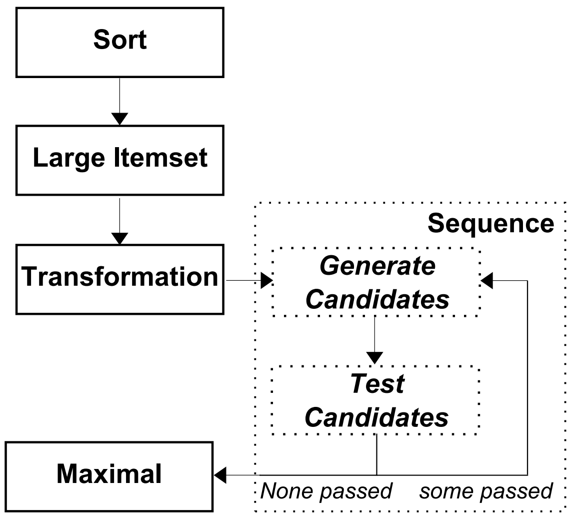


Figure 6.1: A common approach for frequent sequence mining

Within this approach, candidate selection from a large sequence of size k is done as follows:

1. **Join** If two sequences only differ in the last itemset, add that last itemset to the other set.
2. **Prune** All subsets of size k of all candidate itemsets of size $k+1$ should be present in the large sequence set of size k. All candidates failing on this requirement are pruned.

An example can be seen in Table 6.3

Table 6.3: An example of the candidate selection process

Large Sequences	*Candidates (After Join)*	*Candidates (After Prune)*
$\langle 5\ 6 \rangle$	$\langle 5\ 6\ 7 \rangle$	$\langle 6\ 8\ 9 \rangle$
$\langle 5\ 7 \rangle$	$\langle 5\ 7\ 6 \rangle$	
$\langle 6\ 8 \rangle$	$\langle 6\ 8\ 9 \rangle$	
$\langle 6\ 9 \rangle$	$\langle 6\ 9\ 8 \rangle$	
$\langle 8\ 9 \rangle$		

It is clear that a number of steps within this approach are different for the criminal career situation when looking at the different requirements we set forth. The notion of when an itemset is large differs in both situations, occurring either completely in a single transaction as in [2] or occurring in overlapping time frames, per requirement 3. Also, per requirement 3, since the time frame boundaries have no implicit meaning, the notion of time frames is now completely lost, as can be seen in Figure 6.2.

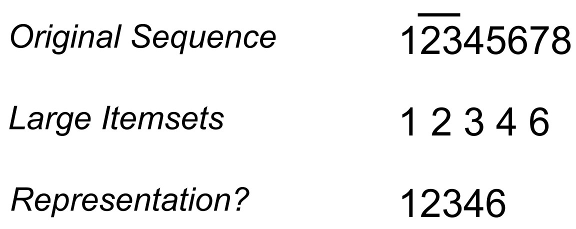


Figure 6.2: A bad representation of a criminal career in phase 3

Through this loss, Figure 6.2 clearly shows that we have unjustly lost sequences 13 and 24 and gained sequence 46, per requirement 2. Therefore, care must be taken in Step 3 to choose a representation that denotes all possible sequences consisting of the frequent itemsets in Phase 2. Depending on the choices made for the transformation phase, we can either keep or change the sequence phase. The options we chose for our approach are discussed in Section 6.3. The first and fifth phase can be kept, regardless of the requirements or choices made for the other steps.

6.3 Approach

As the largest part of the approach described above can still be employed in the search for common subcareers, only a few changes need to be made to the different steps of the approach. They are described below. The main goal of our changes is to retain a straightforward, ad-hoc, human understandable and consecutive approach to the problem, that works reasonably fast, without a very complicated algorithmic foundation.

6.3.1 Changes to the Large Itemset Phase

In the original approach, Large *itemsets* were retrieved in this phase, however, since consecutiveness is required for our cause, the search for crimesets that contain more than one crime could already be seen as a search for subcareers. For example, if the crimeset 12 is Large, this implies that crime 2 immediately follows crime 1. Since the search for sequentiality only starts in phase 4, we should limit our approach to the search for singular items in this case.

An exception to this rule is the situation that both 12 and 21 are Large, indicating that the crimeset $\overline{12}$ is Large as well. However, the inclusion of this set provides no additional gain over the inclusion of separate crimesets 1 and 2; since the inclusion of $\overline{12}$ implies that all its subsets are also frequent per the Apriori-property, we cannot delete extra individuals in the next phase based upon the discovery of a frequent $\overline{12}$. We can therefore safely limit our search for Large itemsets to the search for frequent crimes in the database. Another advantage to this situation is that there is no need to do any Apriori search within the database since a simple count of each crime will yield the required results, greatly reducing the computational complexity of our approach.

Another difficulty that is overcome by the search for single Large crimes alone, is the confusion about the meaning of the statement "$\overline{12}$ is Large". This could either indicate that the summation of occurrences of 12 and 21 is above the threshold, or that both of their occurrences are above the threshold, the first representing a more loose statement and the latter describing a more strict situation. We therefore restrict our search to the retrieval of singular crimes in this phase or simple consecutive subcareers (without overlining) in the next phase.

6.3.2 Changes to the Transformation Phase

Since the crimes within a single time frame can be put in any order, we need to make sure the representation we choose within the transformation phase must retain the ability of searching for all of these possible subcareers. This class of problems is usually known as *trace monoids* [22, 14]. The most natural way of transforming a certain criminal career into a good representation would probably be a directed acyclic graph, that denotes every possible way a certain career can be traversed. Note that, since a time frame is a multiset and crimes can therefore appear more than once in such a time frame, the subgraph that represents this time frame can not be deterministic.

Also, if two crimes appear next to one another after the discarding of another crime (that was not frequent in Step 2), the edge between those two crimes should be tagged to indicate that this edge can not be traversed within the matching process of a single subcareer.

The process of transforming a career is now as follows:

1. **Discard infrequent crimes** Single crimes that do need have a support above the threshold can not be part of any frequent subcareer (Apriori-property). They should therefore be discarded from the graph. A symbol ($) is inserted if this would result in the consecutive paring of crimes that were not consecutive in the original career.

2. **Transform to graph** The career in the form of a string of crime numbers is now transformed into a directed, acyclic graph. The symbol $ is transformed into an "untraversable" edge within a single subcareer. Each time frame with multiple elements is transformed in such a way that a path exists in the graph for every possible subcareer in this time frame. If this time frame is of size *n*, the number of possible paths in the resulting subgraph for this time frame will be *n*!, possibly greatly increasing the computation time. However, the severity of this effect largely depends of the size of these time frames in the actual database, which is investigated in Section 6.4. The directed graph for a time frame can be seen as a tree, since it branches out. Especially when such a tree is very large, it is possible to reduce memory load by specifying multiply occurring subtrees only once and connecting to them from all relevant points in the tree.

3. **Add escape edges** An unfortunate effect that occurs within this approach is the following. Suppose we consider the criminal career $\overline{123456789}$ and would try to match the subcareer 789. The consecutiveness demand requires that 7, 8 and 9 are consecutive so matching should be done at the leaves of the tree, effectively requiring the tree to be "searched" for leaves with the subcareer 78, even though the same subcareer is also present close to the root. As a solution we propose to add escape edges that connect every none leaf directly to the first node after the tree, redirecting the traversion to the so-called "main-path" as quickly as possible. this way a tree search can be prevented, linking the first occurrence of 78 from the example directly to the 9 of the next time frame. This process is illustrated in Figure 6.3, where only the escape edges for the first level have been added as dotted lines. Of course, the second level has escape edges to node *y* as well. The addition of escape edges obviously increases memory load. The effects of this addition are investigated in Section 6.4. Note that an escape edge can only be traversed when the matching of a certain subcareer started at (one of) the root(s) of that subtree, otherwise the consecutiveness requirement could be violated. An example of this violation would be the successful matching of subcareer *x*3*y* (*x* and *y* are only connected through all elements of the tree (1, 2 *and* 3 in this case)).

The process of transforming a career according to this paradigm can be viewed in Figure 6.4. More efficient methods exist for matching a substring in a trace monoid, but

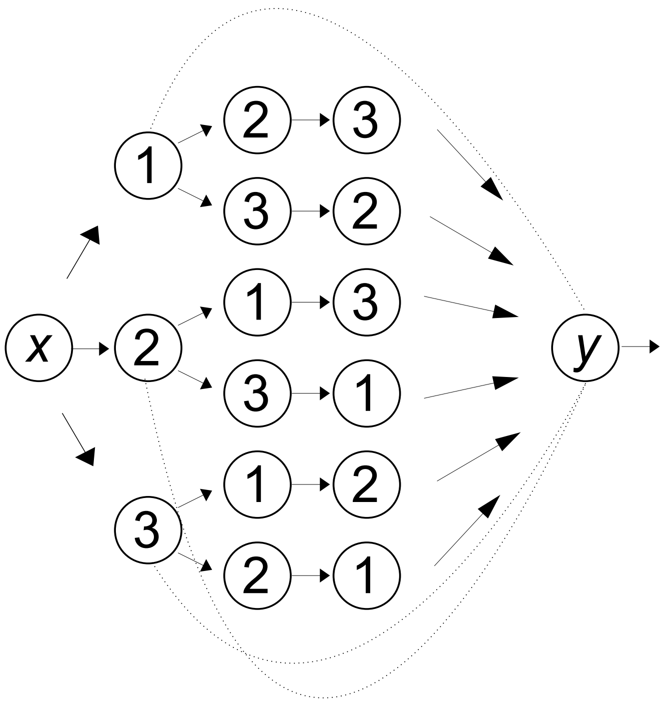


Figure 6.3: The addition of escape edges (dotted) for the first level

they require a reduction of the target string. Since we are matching a large number of completely different substrings, an equal amount of reductions would have to be done. Therefore, it is more efficient, for our approach, to do one transformation that can be easily traversed throughout the entire sequence phase, instead of multiple reductions of the target string, even though the initial calculation takes longer and more memory is required to store the representation.

Using the directed graph representation, there are no major changes needed in the sequence detection phase of Step 4. Although the implementation of graph traversion compared to transaction overlaying can be quite different, the underlying ideas of candidate creation and testing remain the same and the underlying search method unaltered.

6.3.3 Finding Defining Subcareers

Using the mechanism described above, we are now able to detect common subcareers within criminal careers, fulfilling one of our initial goals. The second goal, the detection of subcareers that are only frequent in one specific class, can now be reached easily as well. For this purpose we set two different thresholds, one representing the least amount of occurrences (as a fraction $\mathcal{T}_{\min}$) a subcareer must have to be frequent in a certain class and one representing the maximum number of occurrences (as a fraction $\mathcal{T}_{\max}$) this same

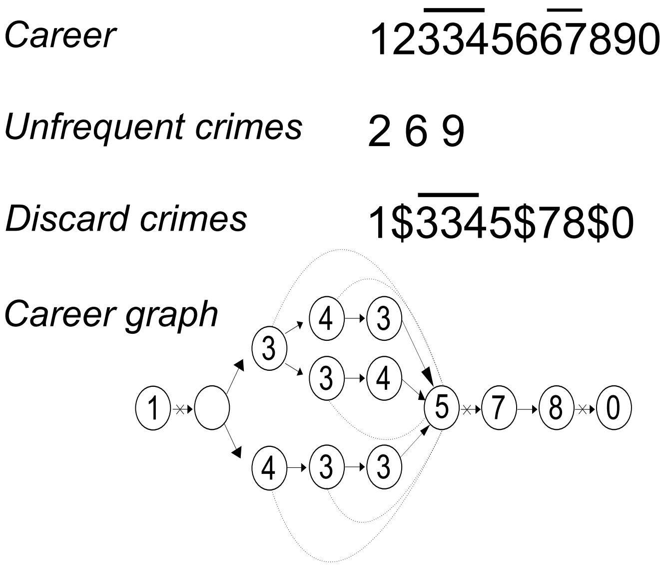


Figure 6.4: The process of transforming a criminal career in Step 3

subcareer can have in all other classes. Note that, counterintuitively, $\mathcal{T}_{\min} > \mathcal{T}_{\max}$. As a rule of thumb, the assumption that both numbers should be the same, the fraction strictly denoting the boundary between a frequent and infrequent status, could be maintained, however, a “grey area” may exist between these thresholds where subcareers are neither frequent nor infrequent. Section 6.4 investigates this relation more closely.

For the detection of these so-called *defining subcareers*, a separate search should be started for all of the 11 classes found in Chapter 5. Consequently there will be 11 different investigations. However, as both frequent and infrequent occurrence should be tested, one would assume all 11 classes would have to be investigated in each of these searches, leading to 11 times the work of the regular search for common subcareers in the entire database.

However, if we employ a three-step approach to this problem, most of this work can be omitted, leading to a very small increase compared to the regular search.

1. **Transformation Phase** All criminal careers in the database are transformed according to the method described above, performing steps 1, 2 and 3 for each class separately. Note that there are possibly crimes that do occur in one class after transformation and do not appear in another. Each career gets assigned its corresponding class number.

2. **Separate Sequence Phase** Each class of criminal careers is searched for common subcareers separately, performing steps 4 and 5 for every class. The results are stored in a Subcareer × Class matrix, where each entry is:

   - 0 Not investigated for this class.

- 1 Frequent in this investigation ($\geq \mathcal{T}_{\min}$).
- 3 Infrequent in this investigation($< \mathcal{T}_{\max}$).

Note that a row is only added when a subcareer is frequent in at least one class. After this phase, the same amount of computer work has been done as in the regular search.

3. **Discarding and Separation Phase** All the rows in the table that have 2 or more entries set to 1 are discarded. All the rows that have exactly one 1 and ten 2's as entries are set aside in a separate table. These represent the defining subcareers we want tot find.

4. **Testing Phase** All remaining rows are tested in all classes where the entry was set to 0. As soon as testing in one of the classes yields a 1 for a specific row, this row is discarded. Within this process, the maximum number of test that are performed is the number of candidate subcareers times 11, if every candidate in the matrix actually yields 2's for every class. Since a very large number of single subcareer tests was already performed in the sequencing phase, only a very small amount of extra computation time is needed for this phase. After this phase, all subcareers that remain are defining subcareers for one of the classes.

6.4 Experimental Results

As in the previous chapters, we tested our approach on the criminal record database (cf. Appendix B), that contains approximately one million offenders and as many criminal careers. Since a large number of the offenders in the list are one-time offenders, these individuals are left out of most of the calculation, greatly reducing the computation time. Furthermore, this class is also left out of the search for common subcareers, since only careers of length 1 are present in this class (except for a small percentage (<0.01%) that was erroneously classified as one-time offender).

As a first test a simple search for common subcareers was performed that yielded the following results:

Table 6.4: Amount of common subcareers discovered per threshold

	50%	40%	30%	20%	10%	5%
Subcareers Found	37	98	243	1,017	5,869	20,017

It appears that 30% is a reasonable threshold, yielding a manageable amount of careers. Figure 6.5 shows clearly how the amount of discovered common subcareers rises in the graph even with an exponential vertical axis. The longest common subcareer we discovered when using a threshold of 30% was of length 7:

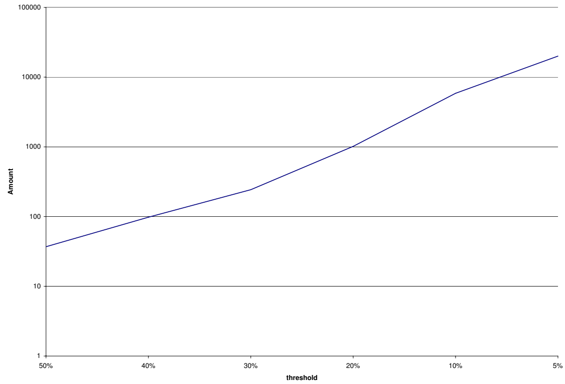


Figure 6.5: The relation between threshold and number of discovered subcareers

Minor Theft → Minor Theft → Minor Theft → Minor Theft → Major Theft → Major Theft → Major Theft

Figures 6.6 and 6.7 show how the discovered common subcareers are distributed over length and how the amount changes with length.

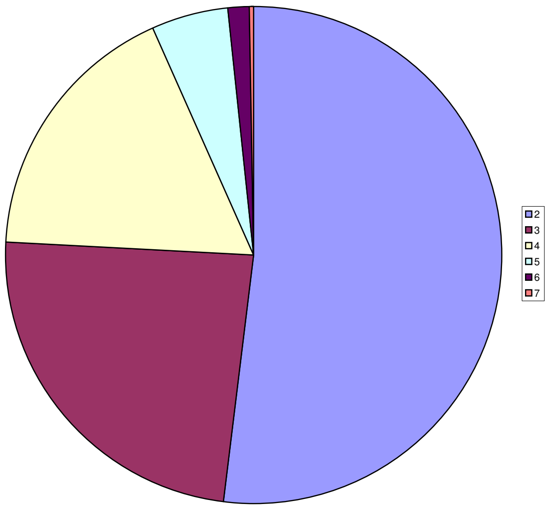


Figure 6.6: The distribution over length

It might be interesting to know how many of the discovered subcareers occur in the different classes. Figure 6.8 shows the division over these classes. It turns out that, when added, they together support 628 common subcareers, being approximately 2.5 times the total amount. This means that an average discovered subcareer appears in 2.6 different classes. The standard deviation in this matter is 1.1.

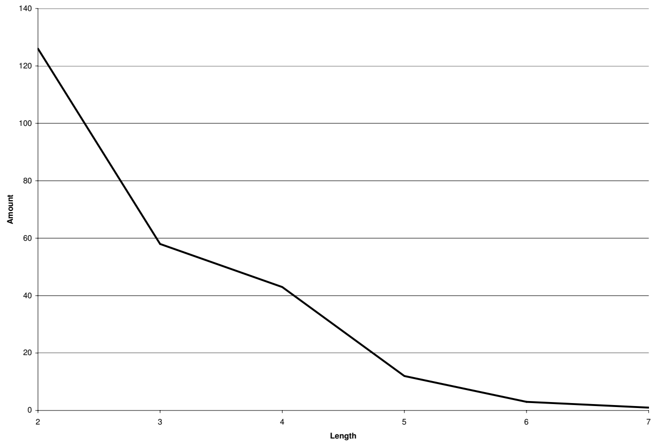


Figure 6.7: The relation between length and number of discovered subcareers

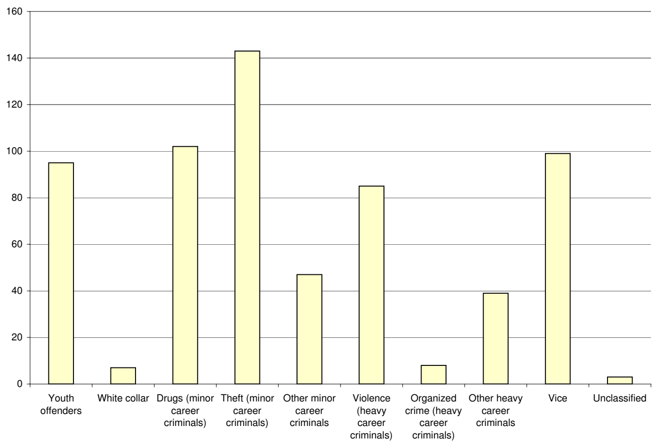


Figure 6.8: Distribution of the results over class

During the creation of the transformation the largest "subtree" (see Section 6.3.2) we encountered contained 6 levels and therefore 720 different paths (threshold: 30%), excluding escape edges, occuring in only one career. This is fortunate, since a slightly higher number would have had some serious effects on the computation time for our representation model. It is also surprising because the maximum amout of crimes in a single time frame in the database is 14. We can therefore state that at least 8 of those crimes were discarded. The maximum amount of paths in a single career, again excluding escape paths, was $6 \times 720 = 4{,}320$, occuring in the same career. Fortunately, both these trees were sufficiently distant from one another in the total career that they were probably not traversed simultaneously.

We investigated how the number of discovered defining subcareers depends on the different values for $\mathcal{T}_{\min}$ and $\mathcal{T}_{\max}$. As it makes no sense to set the threshold for infrequency higher than the threshold for frequency so these were omitted from our tests. Table 6.5 shows the results of batch testing with all possible combinations of $\mathcal{T}_{\min}$ and $\mathcal{T}_{\max}$.

Table 6.5: Amount of common subcareers discovered per threshold

	$\mathcal{T}_{\max}$ = 50%	40%	30%	20%	10%	5%
$\mathcal{T}_{\min}$ = 50%	6	2	0	0	0	0
40%		4	4	3	1	0
30%			112	63	54	3
20%				121	66	4
10%					763	13
5%						812

It is clear that the most results were reached with $\mathcal{T}_{\min}$ at 5 or 10%, only these thresholds were shown to yield too many common subcareers to be called reliable. Again, the best value for $\mathcal{T}_{\min}$ seems to be 30% where even a very strict setting for $\mathcal{T}_{\max}$, 10%, yields a good amount of defining subcareers.

If we choose these settings, we can see how the different defining subcareers are distributed over the different classes. Figure 6.9 shows the results. Clearly, the "Theft" class is best defined by its subcareers, largely outperforming the other classes.

We can now also check the effects of the addition of escape edges in our representation. We redid the experiments both with and without escape edges and checked the change in memory usage and computation time. The results are all averaged over 5 experiments per category. Figure 6.10 shows that the difference in memory load are reasonably low compared to the gain in computation time. Consequently the addition of the escape edges is an asset to our approach.

6.5 Conclusion and Future Directions

In this chapter we described a method that discovers common subcareers from a criminal record database. For this purpose we adapted a well-known algorithm that was more suitable for the retail industry, modifying a number of different phases to suit the search for these frequently occurring subcareers. A representation of the database was chosen that was computationally hard to create but allowed fast searching in the next phase, which led to a better performance overall. Within this representation, we introduced the notion of escape edges that made it easy to find certain types of subcareers that would otherwise have been located very slowly. We also presented a number of results, investigating both

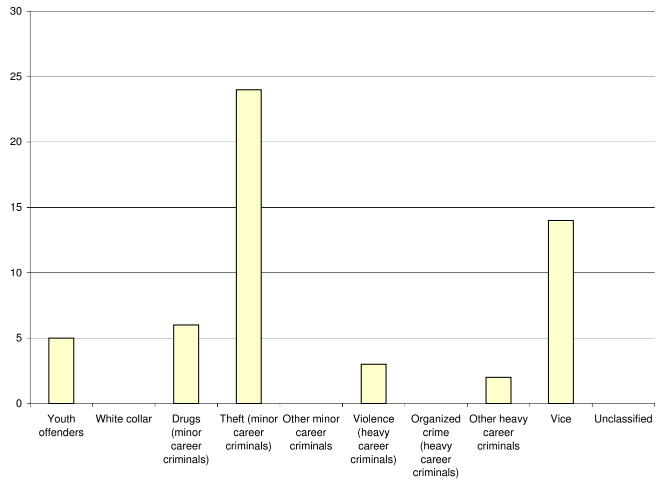


Figure 6.9: Distribution of the defining subcareers over class

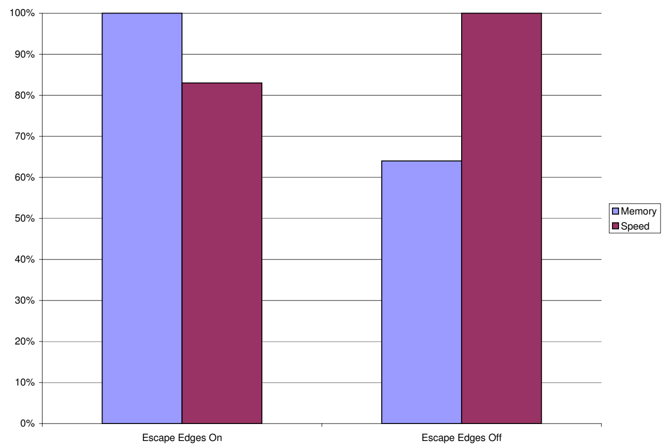


Figure 6.10: Comparison between escape edges off and on as an percentage of the maximum

the usability of our approach on actual data and investigated some of its parameters.

Next to the search for common subcareers, we discovered some subcareers that are frequent in only one class of criminal career (see Chapter 5) and infrequent in all others. These subcareers were discovered through two different thresholds, a minimal threshold to reach frequent status and a maximum threshold denoting the number occurrences this same sequence can have in other classes. Even though we maintained reasonably strict thresholds, a small number of these defining thresholds were discovered.

A possible goal for this research, other than the analysis of crime in general, was to use it as a prediction tool. In this case, a defining subcareers detection for an offender could in theory predict the complete career this individual might develop. However, since only 54 different defining subcareers were discovered, the potential of this research for that purpose is limited. Future research in this area could therefore focus on using the outcome of this result in criminology research to better understand the mechanics of developing criminal careers.

Bibliography for Part I

[1] R. Agrawal, T. Imilienski, and R. Srikant. Mining association rules between sets of items in large databases. In *Proceedings of the ACM SIGMOD Conference on Management of Data*, pages 207–216, 1993.

[2] R. Agrawal and R. Srikant. Mining sequential patterns. In *Proceedings of the Eleventh International Conference on Data Engineering*, pages 3–14, 1995.

[3] S.F. Altschul, W. Gish, W. Miller, E.W. Myers, and D.J. Lipman. A basic local alignment search tool. *Journal of Molecular Biology*, 1990.

[4] A. Blumstein, J. Cohen, J. A. Roth, and C. A. Visher. *Criminal Careers and "Career Criminals"*. The National Academies Press, 1986.

[5] J. Broekens, T. Cocx, and W.A. Kosters. Object-centered interactive multi-dimensional scaling: Ask the expert. In *Proceedings of the Eighteenth Belgium-Netherlands Conference on Artificial Intelligence (BNAIC2006)*, pages 59–66, 2006.

[6] J. Broekens and F.J. Verbeek. Simulation, emotion and information processing: Computational investigations of the regulative role of pleasure in adaptive behavior. In *Proceedings of the Workshop on Modeling Natural Action Selection*, pages 166–173, 2005.

[7] J.S. de Bruin, T.K. Cocx, W.A. Kosters, J.F.J. Laros, and J.N. Kok. Data mining approaches to criminal career analysis. In *Proceedings of the Sixth IEEE International Conference on Data Mining (ICDM 2006)*, pages 171–177. IEEE, 2006.

[8] J.S. de Bruin, T.K. Cocx, W.A. Kosters, J.F.J. Laros, and J.N. Kok. Onto clustering criminal careers. In *Proceedings of the ECML/PKDD 2006 Workshop on Practical Data Mining: Applications, Experiences and Challenges*, pages 92–95, 2006.

[9] A. Califano and I. Rigoutsos. FLASH: A fast look-up algorithm for string homonology. In *Proceedings of the First International Conference on Intelligent Systems for Molecular Biology*, pages 353–359, 1993.

[10] T. K. Cocx and W.A. Kosters. Adapting and visualizing association rule mining systems for law enforcement purposes. In *Proceedings of the Nineteenth*

Belgium-Netherlands Conference on Artificial Intelligence (BNAIC2007), pages 88–95, 2007.

[11] T.K. Cocx and W.A. Kosters. A distance measure for determining similarity between criminal investigations. In *Advances in Data Mining, Proceedings of the Industrial Conference on Data Mining 2006 (ICDM2006)*, volume 4065 of *LNAI*, pages 511–525. Springer, 2006.

[12] T.K. Cocx, W.A. Kosters, and J.F.J. Laros. Temporal extrapolation within a static clustering. In *Foundations of Intelligent Systems, Proceedings of the Seventeenth International Symposium on Methodologies for Intelligent Systems (ISMIS 2008)*, volume 4994 of *LNAI*, pages 189–195. Springer, 2008.

[13] M.L. Davison. *Multidimensional Scaling*. John Wiley and Sons, New York, 1983.

[14] V. Diekert and Y. Métivier. Partial commutation and traces. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages*, pages 457–534. Springer-Verlag, 1997.

[15] A. Dix, J. Finlay, G. D. Abowd, and R. Beale. *Human Computer Interaction*. Prentice-Hall, 3rd edition, 2004.

[16] J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. In *Proceedings of the ACM SIGMOD Conference on Management of Data*, pages 1–12, 2000.

[17] P. Jaccard. Lois de distribution florale dans la zone alpine. *Bull. Soc. Vaud. Sci. Nat.*, 38:69–130, 1902.

[18] W.A. Kosters and J.F.J. Laros. Metrics for mining multisets. In *Proceedings of the Twenty-seventh SGAI International Conference on Artificial Intelligence SGAI2007*, pages 293–303, 2007.

[19] W.A. Kosters and J.F.J. Laros. Visualization on a closed surface. In *Proceedings of the Nineteenth Belgium-Netherlands Conference on Artificial Intelligence (BNAIC2007)*, pages 189–195, 2007.

[20] W.A. Kosters and M.C. van Wezel. Competitive neural networks for customer choice models. In *E-Commerce and Intelligent Methods, Studies in Fuzziness and Soft Computing 105*, pages 41–60. Physica-Verlag, Springer, 2002.

[21] V.I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. *Doklady Akademii Nauk SSSR*, 163(4):845–848, 1965.

[22] A. Mazurkiewicz. Introduction to trace theory. In V. Diekert and G. Rozenberg, editors, *The Book of Traces*, pages 3–41. World Scientific, 1995.

[23] S.B. Needleman and C.D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal on Molecular Biology*, 48:443–453, 1970.

[24] K. Pearson. On lines and planes of closest fit to systems of points in space. *Phil. Mag.*, 2(6):559–572, 1901.

[25] M. Roytberg. A search for common patterns in many sequences. *Computer Applications in the Biosciences*, 8(1):57–64, 1992.

[26] B. Schermer. *Software Agents, Surveillance, and the Right to Privacy: A Legislative Framework for Agent-enabled Surveillance*. PhD thesis, Leiden University, 2007.

[27] T. Smith and M. Waterman. Identification of common molecular subsequences. *Journal on Molecular Biology*, 147:195–197, 1981.

[28] P.J. Stappers, G. Pasman, and P.J.F. Groenen. Exploring databases for taste or inspiration with interactive multi-dimensional scaling. In *Proceedings of IEA2000*, pages 575–578, 2000.

[29] P.-N. Tan, M. Steinbach, and V. Kumar. *Introduction to Data Mining*. Addison-Wesley, 2006.

[30] E. Tejada and R. Minghim. Improved visual clustering of large multi-dimensional data sets. In *Proceedings of the Ninth International Conference on Information Visualisation (IV'05)*, pages 818–825, 2005.

[31] W.S. Torgerson. Multidimensional scaling: I. theory and method. *Psychometrika*, 17(4):401–419, 1952.

[32] M. Vingron and P. Argos. A fast and sensitive multiple sequence alignment algorithm. *Computer Applications in the Biosciences*, 5:115–122, 1989.

[33] J.T.-L. Wang, G.-W. Chirn, T.G. Marr, B. Shapiro, D. Shasha, and K. Zhang. Combinatorial pattern discovery for scientific data: Some preliminary results. In *Proceedings of the ACM SIGMOD Conference on Management of Data*, pages 115–125, 1994.

[34] M. Waterman. *Mathematical Methods for DNA Sequence Analysis*. CRC Press, 1989.

[35] M. Williams and T. Muzner. Steerable, progressive multidimensional scaling. In *Proceedings of the IEEE Symposium on Information Visualization (INFOVIS'04)*, pages 57–64. IEEE, 2004.

[36] S. Wu and U Manber. Fast text searching allowing errors. *Communications of the ACM*, 3(10):83–91, 1992.

[37] M. Zaki, S. Parthasarathy, M. Ogihara, and W. Li. New algorithms for fast discovery of association rules. In *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 283–296, 1997.

Part II

Algorithmic Tools for Tactical Law Enforcement

Chapter 7

Temporal Extrapolation within a Static Visualization

Predicting the behavior of individuals is a core business of policy makers. The creation of large databases has paved the way for automatic analysis and prediction based upon known data. This chapter discusses a new way of predicting the "movement in time" of items through pre-defined classes by analyzing their changing placement within a static preconstructed two-dimensional visualization of pre-clustered individuals. It employs the visualization realized in previous steps within item analysis, rather than performing complex calculations on each attribute of each item. For this purpose we adopt a range of well-known mathematical extrapolation methods that we adapt to fit our need for two-dimensional extrapolation. Usage of the approach on a criminal record database to predict evolvement of criminal careers, shows some promising results.

7.1 Introduction

The ability to predict (customer) behavior or market trends plays a pivotal role in the formation of any policy, both in the commercial and public sector. Prediction of later revenues might safeguard investments in the present. Large corporations invest heavily in this kind of activity to help focus attention on possible events, risks and business opportunities. Such work brings together all available past and current data, as a basis on which to develop reasonable expectations about the future. Ever since the coming of the information age, the procurement of such prognoses is becoming more and more an automated process, extracting and aggregating knowledge from data sources, that are often very large.

Mathematical computer models are frequently used to both describe current and predict future behavior. This branch of data mining, known as *predictive modeling*, provides predictions of future events and may be transparent and readable in for example *rule based systems* and opaque in others such as *neural networks*. In many cases these models are chosen on the basis of *detection theory* [2]. Models often employ a set of classifiers to determine the probability of a certain item belonging to a dataset, like, for example, the probability of a certain email belonging to the subset “spam”. These models employ algorithms like *Naive Bayes*, *K-Nearest Neighbor* or concepts like *Support Vector Machines* [6, 24]. These methods are well suited to the task of predicting certain unknown attributes of an individual by analyzing the available attributes, for example, estimating the groceries an individual will buy by analyzing demographic data. It might, however, also be of interest to predict shopping behavior based upon past buying behavior alone, thus predicting the continuation of a certain sequence of already realized behavior. Examples of this are the prediction of animal behavior when their habitats undergo severe changes, in accordance with already realized changed behavioral patterns, or the prediction of criminal careers based upon earlier felonies.

Sequence learning is arguably the most prevalent form of human and animal learning. Sequences play a pivotal role in classical studies of instrumental conditioning [4], in human skill learning [40], and in human high-level problem solving and reasoning [4]. It is logical that sequence learning is an important component of learning in many task domains of intelligent systems: inference, planning, reasoning, robotics, natural language processing, speech recognition, adaptive control, time series prediction, financial engineering, DNA sequencing, and so on [39]. Our approach aims to augment the currently existing set of mathematical constructs by analyzing the “movement in time” of a certain item through a static visualization of other items. The nature of such a two-dimensional visual representation is far less complicated than the numerical models employed by other methods, but can yield results that are just as powerful. The proposed model can also be added seamlessly to already performed steps in item analysis, like clustering and classification, using their outcome as direct input for its algorithms.

In Section 7.2 we lay out the foundations underpinning our approach, discussed in Section 7.3. In Section 7.4 we discuss the results our method yielded within the area of predicting criminal careers. Section 7.5 concludes this chapter. A lot of effort in this project has gone into the comparison between different types of two-dimensional extrapolation, a field not widely explored in the mathematical world. The main contribution of this chapter is in Section 7.3, where the new insights into temporal sequence prediction are discussed.

7.2 Background

A lot of work has been done in the development of good visualization and strong extrapolation methods that we can resort to within our approach.

7.2.1 Clustering

The goal of *clustering* is the partitioning of a dataset into subsets, that share common characteristics. Proximity within such a dataset is most often defined by some distance measure. It is common practice to visualize such a clustering within the two-dimensional plane, utilizing some form of *Multi-Dimensional Scaling* (MDS) [15] to approximate the correct, multi-dimensional solution. These methods include “associative array” clustering techniques [27], systems guided by human experience [9] and visualizations on different kinds of flat surfaces [25] yielding an image like Figure 7.1. In such an image, the axes do not represent any specific value, but *Principal Component Analysis* [34] could potentially reveal lines that do. Naturally, transforming a multi-dimensional problem to a two-dimensional plane is only possible whilst allowing for some error. Since our approach relies on a static prefabricated clustering, an obvious choice would be to incorporate the method with the smallest error margins in contrast with the method that is the most cost-effective.

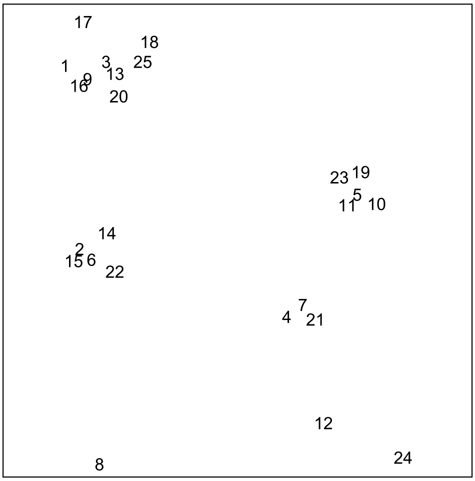


Figure 7.1: A typical clustering of 25 points visualized on a two-dimensional plane

7.2.2 Extrapolation

Extrapolation is the process of constructing new data points outside a discrete set of known data points, i.e., predicting some outcome on a yet unavailable moment (see Figure 7.2). It is closely related to the process of interpolation, which constructs new points between known points and therefore utilizes many of its concepts, although its results are often less reliable.

Interpolation

Interpolation is the method of constructing a function which closely fits a number of known data points and is sometimes referred to as *curve fitting* or *regression analysis*. There are a number of techniques available to interpolate such a function, most of the time resulting in a polynomial of a predefined degree n. Such a polynomial always exactly fits $n+1$ data points, but needs to be approximated if more than $n+1$ points are available.

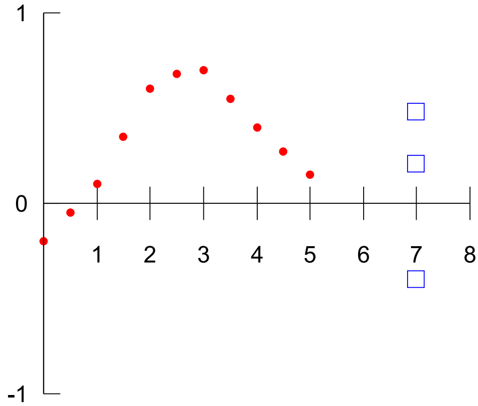


Figure 7.2: The process of trying to determine which of the boxes at time 7 best predicts the outcome based upon the known data points is called extrapolation

In such a case one needs to resort to approximation measures like the *least squares error* method [1].

There are at least two main interpolation methods that are suitable to be incorporated in our approach: *polynomial interpolation* and a *spline*.

Polynomial interpolation tries to find a function

$$p(x) = a_n x^n + a_{n-1} x^{n-1} + \ldots + a_2 x^2 + a_1 x + a_0$$

that satisfies all existing points $p(x_i) = y_i$ for all $i \in \{0, 1, \ldots, n\}$, leading to a system of linear equations in matrix form denoted as:

$$\begin{bmatrix} x_0^n & x_0^{n-1} & x_0^{n-2} & \ldots & x_0 & 1 \\ x_1^n & x_1^{n-1} & x_1^{n-2} & \ldots & x_1 & 1 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ x_n^n & x_n^{n-1} & x_n^{n-2} & \ldots & x_n & 1 \end{bmatrix} \begin{bmatrix} a_n \\ a_{n-1} \\ \vdots \\ a_0 \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{bmatrix},$$

from now on written as $X \cdot A = Y$.

Solving this system leads to the interpolating polynomial $p(x)$ that exactly fits $n + 1$ data points. It is also possible to find a best fit for a polynomial of degree n for $m > n + 1$ data points. For that purpose we project the matrix X on the plane Π_n, the vector space of polynomials with degree n or less, by multiplying both sides with X^T, the transposed matrix of X, leading to the following system of linear equations:

$$X^T \cdot X \cdot A = X^T \cdot Y.$$

Solving this system leads to a best fit polynomial $p(x)$ that best approximates the given data points (see Figure 7.3).

Data points can also be interpolated by specifying a separate polynomial between each couple of data points. This interpolation scheme, called a *spline*, exactly fits the

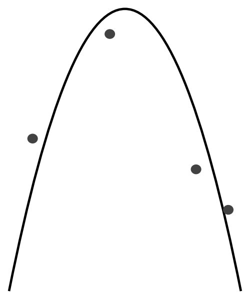


Figure 7.3: A function of degree 2 that best fits 4 data points

derivative of both polynomials ending in the same data point, or *knot*. These polynomials can be of any degree but are usually of degree 3 or *cubic* [5]. Demanding that the second derivatives also match in the knots and specifying the requested derivative in both end points yields $4n$ equations for $4n$ unknowns. Rearranging all these equations according to Bartels et al. [5] leads to a symmetric tridiagonal system, that, when solved, enables us to specify third degree polynomials for both the x and y coordinates between two separate knots, resulting in an interpolation like the graph in Figure 7.4. Due to the liberty this method allows in the placement of the existing data points, this method seems well suited for the task of two-dimensional extrapolation (see Section 7.2.2).

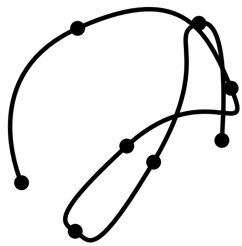


Figure 7.4: An example of a spline

Extrapolation

All interpolation schemes are suitable starting points for the process of extrapolation. It should, however, be noted that higher level polynomials can lead to larger extrapolation errors. This effect is known as the *Runge phenomenon* [35]. Since extrapolation is already much less precise than interpolation, polynomials of degrees higher than 3 are often discouraged.

In most of the cases, it is sufficient to simply continue the fabricated interpolation function after the last existing data point, like for example in Figure 7.2. In the case of the spline, however, a choice can be made to continue the polynomial constructed for the last interval (which can lead to strange artifacts), or extrapolate with a straight line, constructed with the last known derivative of that polynomial. The difference between the two methods is displayed in Figure 7.5.

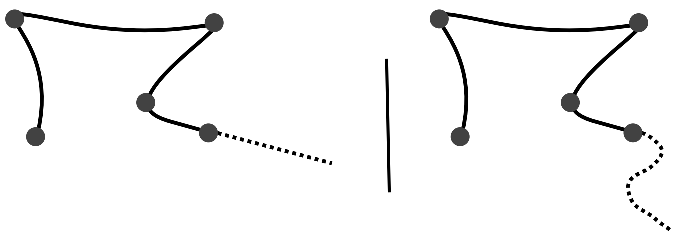


Figure 7.5: The difference between straight line extrapolation (left) and polynomial continuation (right)

Two-Dimensional Extrapolation

In our approach both x and y are coordinates and therefore inherently independent variables. They depend on the current visualization alone and have no intended meaning outside the visualized world. Within our model, they do however depend on the variable t that describes the time passed between sampling points. Because our methods aims to extrapolate the two variables x, y out of one other variable t, we need a form of two-dimensional extrapolation. The standard polynomial function always assumes one independent variable (most often x) and one dependent variable (most often y) and is therefore not very well suited to the required task. However, after rotation and under the assumption that x is in fact the independent variable guiding y, the method can be still be useful. For this scenario we need to establish a rotation that best fits the time ordering to a left-right ordering on the x-axis as displayed in Figure 7.6.

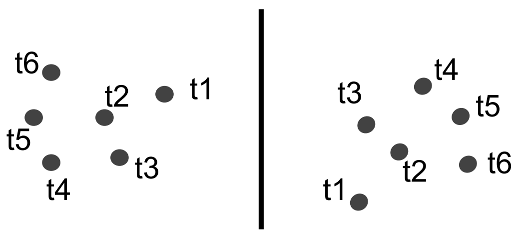


Figure 7.6: Rotation with the best left-right ordering on the x-axis. Note that events 2 and 3 remain in the wrong order

It is also possible to use the polynomial extrapolation for the x and y variables separately and combine them into a linear system, much like spline interpolation, only for the entire domain (referred to as x,y system):

$$p_{x,y}(t) = \begin{cases} x = p_1(t) \\ y = p_2(t) \end{cases}$$

Naturally, the dependence of x and y on t within the spline interpolation scheme makes that method very well suited for the task of two-dimensional extrapolation.

This leaves six methods that are reasonably suited for our approach:

1. Second degree polynomial extrapolation

2. Third degree polynomial extrapolation

3. x,y system with second degree polynomial extrapolation

4. x,y system with third degree polynomial extrapolation

5. Spline extrapolation with straight line continuation

6. Spline extrapolation with polynomial continuation

7.3 Approach

The number of attributes describing each item in a database can be quite large. Taking all this information into account when extrapolating sequences of behavior through time can therefore be quite a hassle. Since this information is inherently present in a visualization, we can theoretically narrow the information load down to two attributes (x and y) per item whilst retaining some of the accuracy. We designed the stepwise strategy in Figure 7.7 for our new approach.

7.3.1 Distance Matrix and Static Visualization

The data used as reference within our approach is represented by a square distance matrix of size $q \times q$ describing the proximity between all q items. These items are considered to be complete in the sense that their data is fully known beforehand. The number of items q should be large enough to at least provide enough reference material on which to base the extrapolation.

These items are clustered and visualized according to some MDS technique resulting into a two-dimensional plane with dots representing our reference items, see, e.g., Figure 7.1. This step in the approach is done only once so the focus should be on the quality of the visualization and clustering instead of the computational complexity. From this point on this visualization is considered to be static and describing the universal domain these items live in. Note that all offenders, with a career we consider to be finished are present in this visualization.

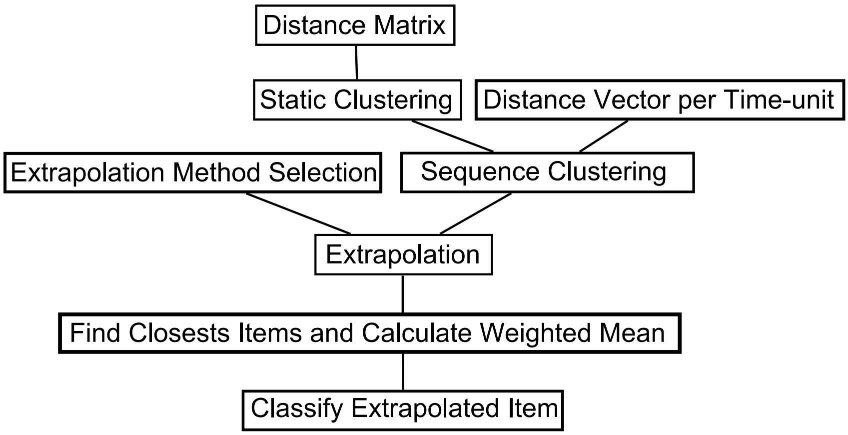


Figure 7.7: Stepwise approach

7.3.2 Distance Vector Time Frame and Sequence Clustering

Analysis of the behavior of new items should start with the calculation of the values for each time frame t. These frames are supposed to be cumulative, meaning that they contain all the item's *baggage* up to the specified moment. Using the same distance measure that was used to create the initial distance matrix, the *distance vector per time frame* can now be calculated. This should be done for all t time frames, resulting in t vectors of size q.

These vectors can now be visualized in the previously created clustering on the correct place using the same visualization technique, while leaving the original reference points in their exact locations. The chosen visualization method should naturally allow for incremental placement of individual items, e.g., as in [27]. These new data points within the visualization will be used to extrapolate the items behavior through the static visualization. because of accuracy reasons, we will only consider items for which three or more time frames are already known. Note that the first time frame will be placed in the cloud of one-time offenders with a very high probability.

7.3.3 Extrapolation

After selecting the best extrapolation scheme for our type of data our method creates a function that extrapolates item behavior. For the same data the different schemes can yield different results as illustrated in Figure 7.8, so care should be taken to select the right type of extrapolation for the data under consideration.

One advantage of this approach is that the extrapolation or prediction is immediately visualized to the end-user rather than presenting him or her with a large amount of numerical data. If the user is familiar with the data under consideration, he/she can analyze the prediction in an eye blink. Augmenting the system with a click and point interface would enable the end-user to use the prediction as a starting point for further research.

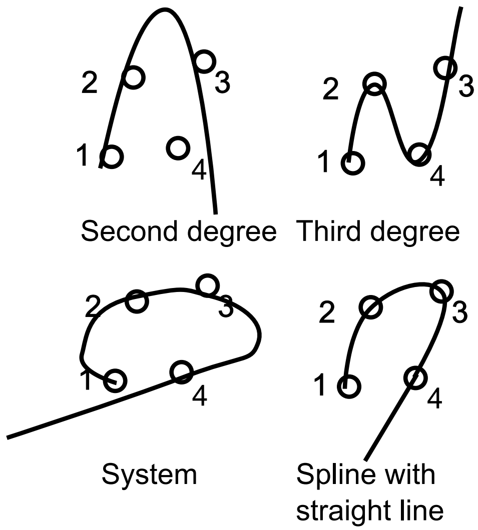


Figure 7.8: Different extrapolation methods yield very different results

7.3.4 Final Steps

In most cases it is desirable to predict which class the item under consideration might belong to in the future. In that case it is important to retrieve further information from some of the reference items and assign future attribute values and a future class to the item.

A first step would be to select a number of reference items *r* closest to the extrapolation curve. This can easily be done by evaluating the geometric distance of all reference points to the line and selecting *r* items with the smallest distance. This process can be seen in Figure 7.9.

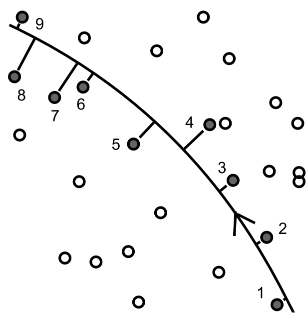


Figure 7.9: Selecting points with the shortest distance to the extrapolation line

Each of these points gets a number i assigned based upon their respective distance to the last known data point of the extrapolated item (see Figure 7.9). Because of the declining confidence of the prediction further away from that last point this number represents the amount of influence this item has on the extrapolated classification. We are now able to calculate the weighted mean value for each future attribute j according to the following formula:

$$Attrib_j(new) = \frac{2}{r+1} \cdot \sum_{i=1}^{r} (r - i + 1) Attrib_j(i)$$

Here, $Attrib_j(i)$ denotes the value of the j^{th} numbered attribute of i.

The extrapolated item can now be visualized together with the clustering according to its future attributes and be classified accordingly.

7.4 Experimental Results

The detection, analysis, progression and prediction of criminal careers is an important part of automated law enforcement analysis [10, 11]. Our approach of temporal extrapolation was tested on the criminal record database (cf. Appendix B), containing approximately one million offenders and their respective crimes

As a first step we clustered 1000 criminals on their criminal careers, i.e., all the crimes they committed throughout their careers. In this test-case r will be set to 30. We employed a ten-fold cross validation technique within this group using all of the different extrapolation methods in this static visualization and compared them with each other and standard extrapolation on each of the attributes (methods 7 and 8). For each item in the test set we only consider the first 4 time periods. The accuracy is described as the percentage of individuals that was correctly classified (cf. Chapter 5, compared to the total amount of individuals under observation. The results are presented in Table 7.1, where *time factor* represents how many times longer the method took to complete its calculation than the fastest method under consideration.

Although the calculation time needed for visual extrapolation is much less than that of regular methods, the accuracy is very comparable. For this database the best result is still a regular second degree extrapolation but its accuracy is just marginally higher than that of the spline extrapolation with a straight line, where its computation complexity is much higher. The simpler x,y system with third degree extrapolation has got a very low runtime complexity but still manages to reach an accuracy that is only 1.5 percentage points lower than the best performing method.

7.5 Conclusion and Future Directions

In this chapter we demonstrated the applicability of temporal extrapolation by using the prefabricated visualization of a clustering of reference items. This method assumes that the visualization of a clustering inherently contains a certain truth value that can yield

Table 7.1: Results of Static Visualization Extrapolation for the analysis of Criminal Careers

	method	*time factor*	*accuracy*
1	Second degree polynomial extrapolation	1	79.1%
2	Third degree polynomial extrapolation	1.1	79.3%
3	*x*,*y* system with second degree polynomial extrap.	1.9	81.5%
4	*x*,*y* system with third degree polynomial extrap.	2.1	87.5%
5	Spline extrapolation with straight line continuation	13.4	88.7%
6	Spline extrapolation with polynomial continuation	13.4	79.6%
7	Regular second degree attribute extrapolation	314.8	89.0%
8	Regular third degree attribute extrapolation	344.6	82.3%

results just as powerful as standard sequence extrapolation techniques while reducing the runtime complexity by using only two information units per reference item, the x- and y-coordinates. We demonstrated a number of extrapolation techniques that are suitable for the extrapolation of the temporal movement of a certain item through this visualization and employed these methods to come to a prediction of the future development of this item's behavior. Our methods were tested within the arena of criminal career analysis, where it was assigned the task to predict the future of unfolding criminal careers.

We showed that our approach largely outperforms standard prediction methods in the sense of computational complexity, without a loss in accuracy larger than 1 percentage point. On top of that, the visual nature of our method enables the analyst of the data to immediately continue his/her research since the prediction results are easily displayed within a point and click interface, rather than presenting him with unnecessary detailed numerical results.

It is important to perform the test described in Section 7.4 for each new type of data that is subject to this approach. Different types of data might well be more susceptible to errors in one of the extrapolation methods and less in others.

Future research will aim at reaching even higher accuracy values by improving the selection of reference items close to the extrapolation line. Incorporation of this approach in ongoing research in the area of criminal career analysis should reveal the power and use of this approach in law enforcement reality and should provide a plethora of improvements to the method.

Any concerns about privacy and judicial applicability on the usage of the methods described here are discussed in Appendix A.

Chapter 8

An Early Warning System for the Prediction of Criminal Careers

Dismantling networks of career criminals is one of the focus points of modern police forces. A key factor within this area of law enforcement is the accumulation of delinquents at the bottom of the criminal hierarchy. A deployed early warning system could benefit the cause by supplying an automated alarm after every apprehension, sounding when this perpetrator is likely to become a career criminal. Such a system can easily be built upon existing, strategic, analysis already performed at headquarters. We propose a tool that superimposes a two-dimensional extrapolation on a static visualization, that describes the movement in time of an offender through the criminal spectrum. Using this extrapolation, possible future attributes are calculated and the criminal is classified accordingly. If the predicted class falls within the danger category, the system could notify police officials. We outline the implementation of such a tool and highlight test results on a criminal record database. We also touch upon the issue of estimating the reliability of a single, individual criminal career prediction.

8.1 Introduction

The dismantlement of crime syndicates ranks seventh in the current list of top priorities of the FBI [21]. As the growth of crime syndicates starts at the bottom layer of the criminal hierarchy, which is most often shaded to law enforcement agencies, a tool that makes educated guesses about people who are most likely to enter such organizations can be a valuable asset. This chapter discusses a new tool that attempts to predict the continuation of individual criminal careers: the criminal activities that a single individual exhibits throughout his or her life. Analysis of a criminal record database (described in Appendix B) leads to a clustering of careers, that yields important information for the formation of strategic policies [13]. Furthermore, the visualization can serve as a basis on which to track the movement in time of a certain perpetrator. A plotted line though the first few years of such a career could potentially be extended and a future class could be assigned to such an individual. Integration of this toolset into police software enables the automatic prediction of a criminal career each time a new entry for an offender is submitted. If the calculated career then falls within a preconfigured set of danger categories, an early warning will be sent to police officers in charge of for example organized crime, and action can be taken accordingly. In this chapter, we discuss the challenges in career prediction and the specific problems that arise with the implementation of software that transfers higher levels of knowledge discovery to prediction of individual cases.

8.2 Background

A new way of predicting the "movement in time" of items through predefined classes by analyzing their changing placement within a static, preconstructed two-dimensional visualization of other individuals was discussed in Chapter 7, [14]. It employs the visualization realized in previous steps within item analysis, rather than performing complex calculations on each attribute of each item. For this purpose a range of well-known mathematical extrapolation methods was adopted that were adapted to fit the need for two-dimensional extrapolation.

As a first step in this paradigm, the individual sequence to be extrapolated is selected and the sequence up to that moment is calculated for each time frame. Then, the distance between all time frames and all other sequences already in the visualization is calculated. Each time frame is then clustered in the original clustering, leaving all the existing elements in their original location. Note that this requires an iterative visualization and clustering method, like for example a randomized push-and pull-algorithm, rather than a Multi-Dimensional Scaling technique that calculates the entire visualization in a single pass. The resulting coordinates for each time frame can now be utilized by an extrapolation scheme.

The visual extrapolation paradigm has two distinguished advantages. On one hand, the results can immediately be visualized to the end-user, also enabling the user to derive how the results were reached in the first place, on the other hand, the computational complexity is very low, requiring only a few distances to be calculated and only a few elements to be plotted within an existing visualization. Note that the calculated x and y coordinates have no intended meaning; they serve only to give an idea where the item under consideration will be displayed relative to existing elements (initially also positioned on an arbitrary location).

The approach offers several different extrapolation schemes that are suitable for usage within a plane: second or third degree polynomial extrapolation, an x,y system with second or third degree polynomial extrapolation or spline extrapolation with straight line or polynomial continuation. Depending on the task domain the one with the best results should be selected. More information on visual extrapolation and the different extrapolation schemes can be found in Chapter 7.

8.3 Approach

The incorporation of a prediction tool into regular, police software comes with some time constraints; the early warning system should obviously not interfere with regular daily operations. Hence, the computational complexity of a prognosis tool should be minimal. Standard mathematical extrapolation methods that, for example, extrapolate every attribute separately, have difficulties complying with this demand. Next to that fact, a series of 0's will always be extrapolated to 0 by standard approaches. Some crimes, however, tend to have the property that they are far more likely to be committed after a few years of criminal activity, effectively creating a series of 0's that needs to be extrapolated by a number other than 0. This effectively renders standard extrapolation inapplicable here. Using the visualization as a depiction of domain knowledge, this problem can be dealt with effectively. We therefore resort to a more knowledge discovery oriented approach, specifically the two-dimensional extrapolation of visualization results mentioned above. As was shown in Chapter 7, the power of a clustering visualization resulting from career comparison can easily be used to reach accurate results without an abundance of computations. Using the temporal extrapolation method, only the *coordinates* of the careers in the visualization are used for the largest part of the algorithm. In Figure 8.1 we show the steps we take to come from the mentioned visualization to a decision on issuing a warning. Each step is described below.

Within this figure, only the boxed steps are taken every time the method is used to determine the development of a single career. The other steps are taken beforehand (the visualization and underlying clustering) or are predetermined by research (selection of extrapolation method, clustering reduction and reference points) or by domain experts (selection of danger categories).

As different steps of our approach take place in spaces with different dimensions, an overview is provided in Figure 8.2.

In this Figure, all steps within the high dimensional space are performed in the processes described in Chapters 4 and 5. These chapters end with the construction of a two-dimensional visualization (obviously a reduction to two-dimensional space) and a classification method that incorporates both the visualization and the raw high dimensional data from the database. The extrapolation methods in Chapter 7 built upon these purely two-dimensional results and combined them with the raw data to predict a future number of crimes. That information is then combined with the classification system in combined space to predict a future class.

8.3.1 Clustering Reduction and Extrapolation Selection

Although the speed of the temporal extrapolation scheme is high, the accuracy of the results can vary with the selection of a specific extrapolation method. It is important to determine the optimal option for extrapolation through field testing, which is explored in the experiments in Section 8.4.

Given the size of a typical criminal record database, ranging in the millions, a significant gain in speed could be realized by reducing the amount of offenders within the

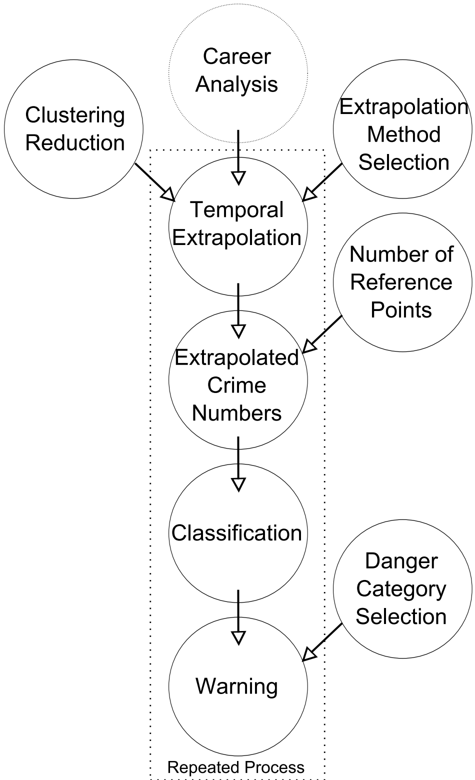


Figure 8.1: Approach for criminal career prediction

clustering. Naturally, care must be taken to realize this decrease without sacrificing the descriptiveness of the clustering itself, for example because certain clusters might lose enough "members" to cause a substantial reduction in "attraction" to newly entered individuals. Within our approach, the reduction is realized using a top down approach (as seen in Figure 8.3), that deletes items from a y-coordinate sorted list, keeping only every tenth individual.

This will reduce the amount of individuals with a factor 10, retaining a higher similarity with the original clustering than what would be the case if just the first tenth of the original database was used. The rationale behind this is that using this method, the same amount of individuals will be removed from every "height" in the image, preserving the shape of the image as much as possible. A strong argument against simple database removal is the fact that the database could be sorted in many ways, both implicitly and explicitly, without the user's knowledge. Therefore, removal of specific parts of the database could have unintended effects on the outcome, creating a "polluted" clustering. If necessary this process can be repeated to create even smaller clustering sizes. The effects of this reduction are analyzed and discussed in Section 8.4.

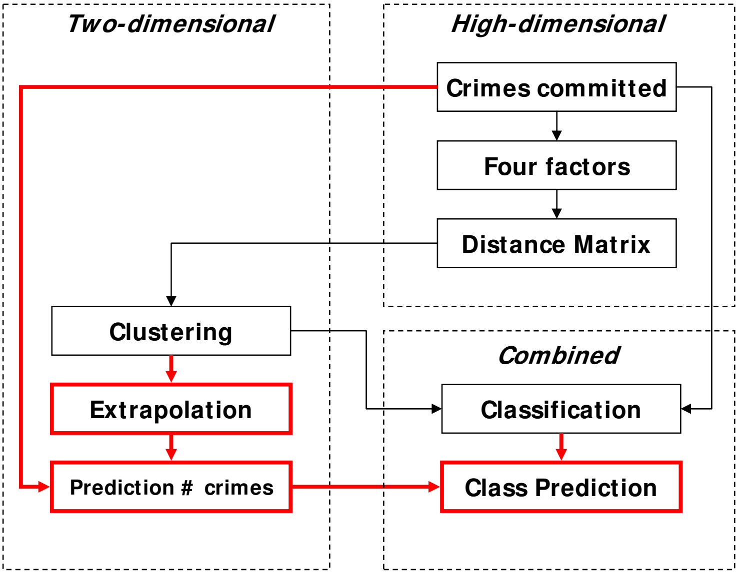


Figure 8.2: Different prediction steps take place in different dimensions

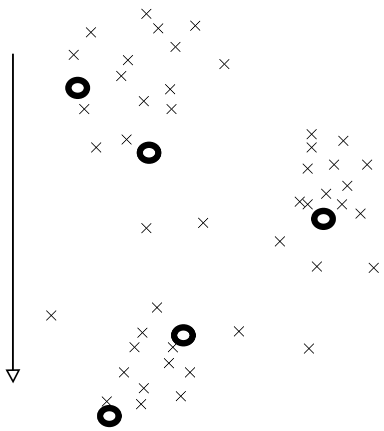


Figure 8.3: Example for clustering reduction, shown in a visualization; only the circles are kept

8.3.2 Further Steps

A typical temporal extrapolation for an emerging minor career criminal is found in Figure 8.4. Here each cross represents a year of known criminal activity and the dotted line denotes the expected continuation of the sequence or criminal career.

Domain experts can easily scan such images and conclude what class of criminal career this individual will probably belong to (minor career criminal in the case of the situation in Figure 8.4). If incorporation of the prediction is wanted, however, it is necessary to classify the offender under observation. Hence, we need to automatically calculate its attributes or the number of different crimes in his or her future. This can be accomplished by selecting a number of *reference points* close to the extrapolated line, averaging over their respective crime numbers to reach the expected crime data for the current offender. It was suggested in Chapter 7 that reference points closest to the last known year receive a higher weight in this process, following

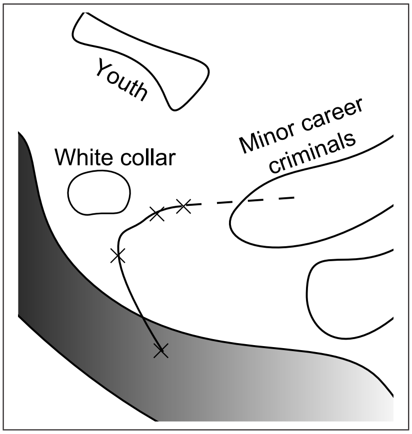


Figure 8.4: Example of temporal extrapolation for criminal careers

$$Attrib_j(new) = \frac{2}{r+1} \cdot \sum_{i=1}^{r} (r - i + 1) Attrib_j(i),$$

where r is the amount of reference points and j is one of the crime types. This process is illustrated in Figure 8.5.

Of course, the number of reference points to use is a matter of accuracy versus time complexity. Looking up a large number of reference points in a database can be very time consuming, but selecting a small amount can cause accuracy to drop greatly. Selection of the right number of reference points can therefore contribute to successful implementation of this tool and is discussed in Section 8.4.

Now that the possible future crime numbers are calculated, the individual can easily be classified in one of the categories. Domain experts can select which categories to monitor based upon their domain knowledge, the specific needs of their own district or the specific tasks of policing they are involved in. A warning can be issued on their computer every time a new individual falls within one of the selected danger categories.

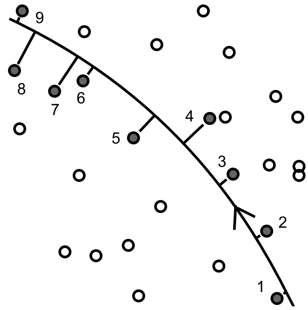


Figure 8.5: Selecting points with the shortest distance to the extrapolation line

8.4 Experimental Results

A number of experiments was performed to both reveal acceptable values for the needed parameters and test the validity of the approach as a whole. For this purpose the criminal record database described in Appendix B, was used to run the algorithm with a number of different settings. It contains approximately one million offenders and their respective crimes, of which 10% could be said to have finished their careers (no reported crimes for the last 10 years). Although this selection method is coarse, people can be incarcerated or they were simply not caught, it can still be used as a validation group.

Of these 10% the career length is distributed as is displayed in Figure 8.6.

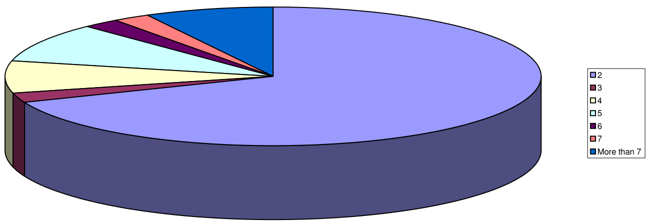


Figure 8.6: Career length distribution within finished career subset

As a first step we clustered n criminals on their (in most cases) finished criminal careers, i.e., all the crimes they committed throughout their careers. In our first test, the number of reference points was set to $r = 100$.

A ten-fold cross validation was used to calculate the accuracy of a certain method: for each fold, one tenth of the population was used as a test group, where the other careers were used in the creation of the clustering. All the careers in the population were "cut off" after 3 or 4 years. For each of those careers, the future crime number was predicted and compared with the actual end-values of their careers. The accuracy for one career

Table 8.1: Time complexity and accuracy comparison between different methods and clustering sizes using r = 100 reference points

Clustering Size (n)	**1,000,000 (all)**	**100,000**	**10,000**	**1,000**
Second degree polynomial	961	98	10.4	1.0
	79.1%	77.9%	75.7%	61.3%
Third degree polynomial	965	101	10.5	1.1
	79.3%	78.2%	75.8%	62.0%
x,y **system (second degree)**	971	104	11.3	1.9
	81.5%	81.1%	79.9%	66.6%
x,y **system (third degree)**	973	105	11.7	2.1
	87.5%	87.3%	86.3%	74.4%
Spline (straight line)	982	113	**22.0**	13.4
	88.7%	88.2%	**87.3%**	74.3%
Spline (polynomial)	983	114	22.1	13.4
	79.6%	78.4%	76.9%	63.7%

prediction will then be described by the average similarity between all 50 predicted and actual crime numbers. The accuracy of the method using these settings is then described by the mean of all averages.

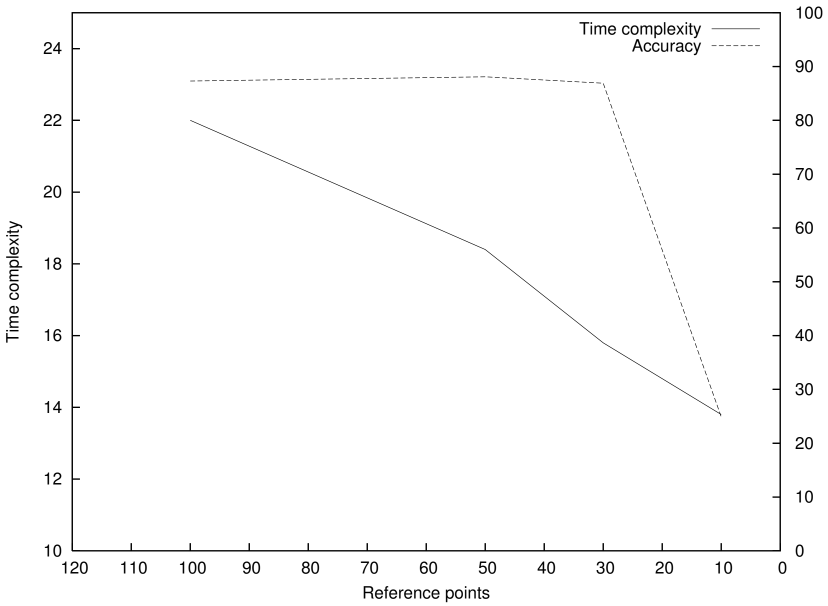
For all methods, a time factor was also calculated, This time factor represents how much time was consumed, using this method within this clustering size, relative to the fastest method-size combination (which is set to 1). On standard equipment this method needed an average of 60 ms.

The results are presented in Table 8.1, where the top box of every cell describes the time factor and the bottom box contains the calculated accuracy.

From the above presented results, the conclusion can be drawn that the x,y system with a third degree polynomial and the spline with straight line extrapolation largely outperform the other methods, especially when the amount of careers in the clustering decreases. The spline performs slightly better than the system methodology.

The decrease in clustering size appears to have a minor effect on accuracy, lowering it only marginally while reducing the clustering size with a factor 100. A steep drop, however, occurs when the size is lowered to 10,000 careers. Apparently, the quality of the clustering reaches a critical low, to make a reliable prediction.

Given the time demands put on this application, the best choice would be to overlay a straight line spline extrapolation on a 10,000 size clustering (bolded option in Table 8.1). The accuracy of this option can be considered high and provides a solid foundation for incorporation, while allowing for fast calculation (approximately 1.3 seconds).



	100	50	30	10
Time complexity	22.0	18.4	15.8	13.8
Accuracy	87.3%	88.1%	86.9%	24.9%

Figure 8.7: The relation between accuracy and time complexity when reducing the number of reference points

Potentially, an even greater gain in time can be reached by reducing the number of reference points, thus reducing calculation of the different averages. Figure 8.7 describes the effects on accuracy and time complexity of reference point reduction, using the optimal solution described above.

Again, the reduction in information does not (necessarily) lead to decrease in quality. Reducing the number of reference points to 30, slightly lowers the accuracy with only 0.4 percentage points. Furthermore, a reduction to 50 leads to an increase of 0.8 percentage points, probably because the selection method selects careers that are simply too far away from the extrapolated line to contribute positively to the calculation of crime numbers. A steep decline can be seen with the reduction of reference points below 30. Depending on the need for speed-up or the quality of the prediction any number between approximately 50 and 30 can be selected.

It may also be of interest to see the influence of the amount of years of criminal activity that are already known on the result. In the example above, either 3 or 4 years were selected. In Figure 8.8 we show how the accuracy depends on the availability of realized

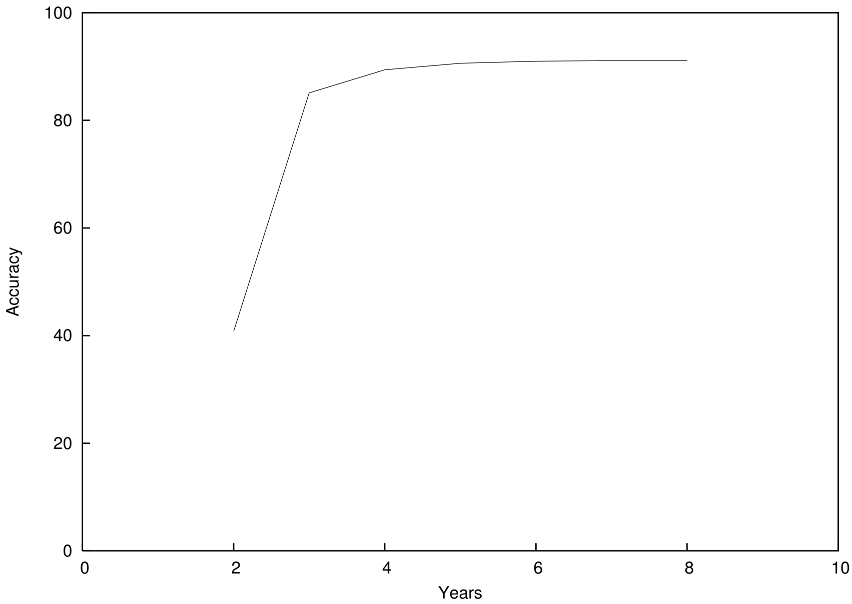


Figure 8.8: Accuracy as a function of known years

behavior. For this experiment we used the straight line spline extrapolation, a clustering of size 10,000 and 50 reference points. Only individuals with more than 10 years of activity in total were selected for testing the extrapolation accuracy in this experiment.

As can clearly be observed in the graph, a career of which less than 3 years of activity are already recorded can not be predicted accurately. However, the results cease to improve with the addition of more than 5 years of activity. As could be expected (2 points are best extrapolated by a straight line, which in most cases does not validly predict a career), prediction efforts should start only for criminals who are active for more than 2 years, in which case an accuracy of 88% can be reached within 1.2 seconds.

It might be possible that offenders are already classified in their respective end-classes after examination of the data that is already known. If this is the case in a large portion of the database for the amount of reference years used within our approach, the necessity of extrapolation and the expressiveness of its accuracy would greatly decrease. However, Figure 8.9 shows that only a reasonably small percentage of offenders have reached their respective end-classes after the amount of reference years we used to reach the results described above.

Combined with the results displayed in Figure 8.8, the added value by using our prediction method over the simple assumption that criminal careers do not change after a set number of reference years is displayed in Figure 8.10.

Within this figure, the top portion of the graph denotes the gain that is reached using our method specified per amount of reference years used. It is clear that using two-

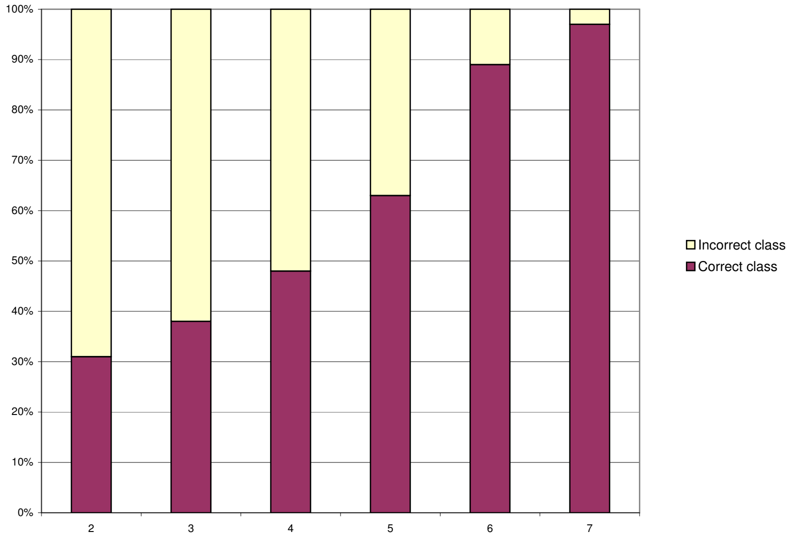


Figure 8.9: Accuracy without prediction after known years

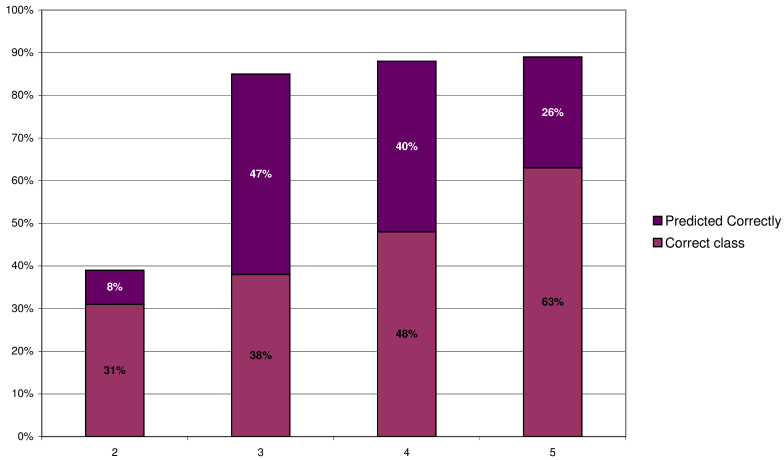


Figure 8.10: The additional accuracy reached through prediction

dimensional extrapolation, the prediction of careers can be greatly enhanced within the [2 . . . 5] reference year domain, especially when using 3 or 4 reference years. When 6 years of activity are known the added value of our approach approaches zero and after 7 years of activity the prediction yielded by the extrapolation becomes less accurate than the actual class the offender under consideration was already placed in.

Summarizing all of the above results, a two years ahead advantage in the prediction of criminal careers can be reached with two-dimensional straight line spline extrapolation, using 3 or 4 reference years and between 30 and 50 reference points.

8.4.1 Accuracy of a Single Prediction

Obviously, the overall accuracy of 89% that can be reached is not representative for the extrapolation of an individual sequence of events, meaning that an individual career can be predicted with an 89% certainty, but that results may vary greatly, depending on the type of offences, the amount of crimes or the shape of the extrapolation line. Therefore, further insights into the accuracy of a single prediction are necessary when investigating a single criminal career. In such a case, a percentage denoting the chance that this specific prediction is correct is as a measure or reliability that law enforcers can use.

For this purpose, we propose to use the shape of the extrapolation curve as an indication for the reliability of its prediction, assuming that a curve with a variance in its direction has a less reliable outcome than a curve that slowly changes towards its end-class. Within this effort, we will be using the modification of the derivative of the curve as an indication for directional changes. Since we are using the straight line spline extrapolation, only the interpolated part of the curve needs to be observed (the derivative does not change after the last known data point), providing us with a fixed domain to be investigated. This allows for a change detection based upon small changes in t, that, knowing its start- and end values, can be set at a static interval. A natural way to denote the modification of the derivatives is in degrees of the smallest angle between the old and the new tangent, which should yield the correct result as long as Δt is small enough and the curves do not change direction too suddenly.

Figure 8.11 describes the mathematical situation when calculating the distance between two tangents. Figure 8.12 shows a situation where the calculation is somewhat different.

The two lines in these figures represent the tangents, angle $\angle xy$ is the target angle to be calculated and $\angle x$ and $\angle y$ are the angles between the tangents and the horizontal assisting line. Using this situation,

$$\angle xy = \begin{cases} \tan^{-1}\max(|\alpha_x|, |\alpha_y|) - \tan^{-1}\min(|\alpha_x|, |\alpha_y|) & \alpha_x\alpha_y \geq 0 \\ \tan^{-1}|\alpha_x| + \tan^{-1}|\alpha_{y|} & \alpha_x\alpha_y < 0 \end{cases},$$

where α_x and α_y are the slopes of tangents x and y, respectively. Within our experiments we used three reference years and set Δt on 0.005 ($t \in [0, 3]$), having 600 intervals, each with its own $\angle xy_t$. The directional change of interpolation curve c, Δ_c can now be calculated as follows:

$$\Delta_c = \sum_{t=0}^{600} \angle xy_{\frac{t}{600}}$$

In order to evaluate the fitness of this function as an indicator for prediction reliability a Δ_c was added to every possible prediction. As can be seen in Figure 8.13, Δ_c is

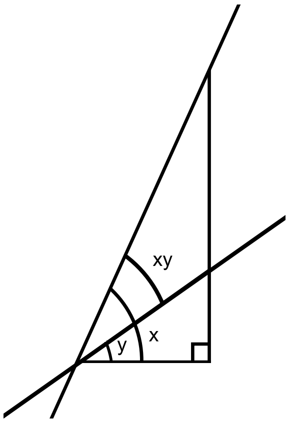


Figure 8.11: The mathematical situation when calculating the angle between two tangents

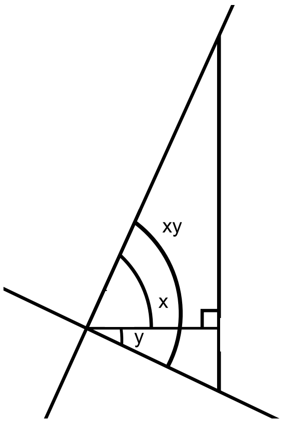


Figure 8.12: Another possible calculation of the angle between two tangents

approximately normally distributed with a mean around 90 and a standard deviation of about 50 degrees. Each occurrence is calculated for a 10 degree interval and is specified in thousands.

According to our previous results, the average of 89% accuracy should be reached at approximately 90 degrees curvature. A closer observation of these 10 degree intervals should reveal if there is any connection between Δ_c and the accuracy of the prediction. If we calculate the percentage of correctly predicted offender classes for each interval separately, and add a trend line, a relation could be established together with its average error. Figure 8.14 shows the result of this investigation.

In this figure, the vertical axis is the accuracy and the horizontal axis is Δ_c. As the

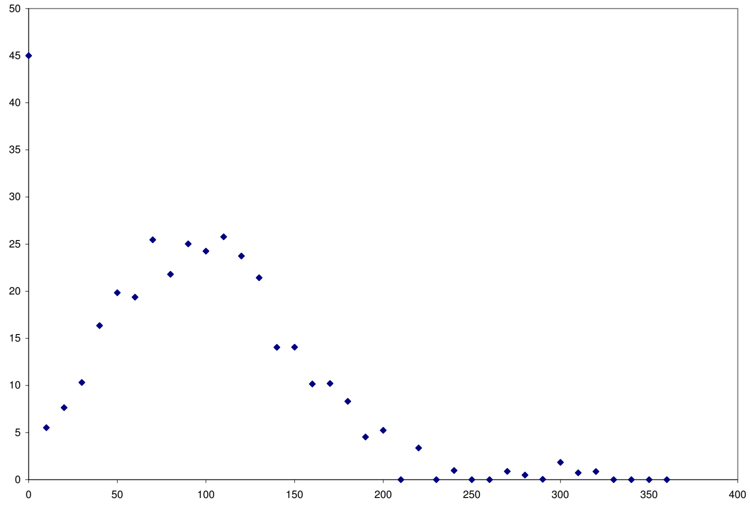


Figure 8.13: Scatterplot of complete population test of Δ_c

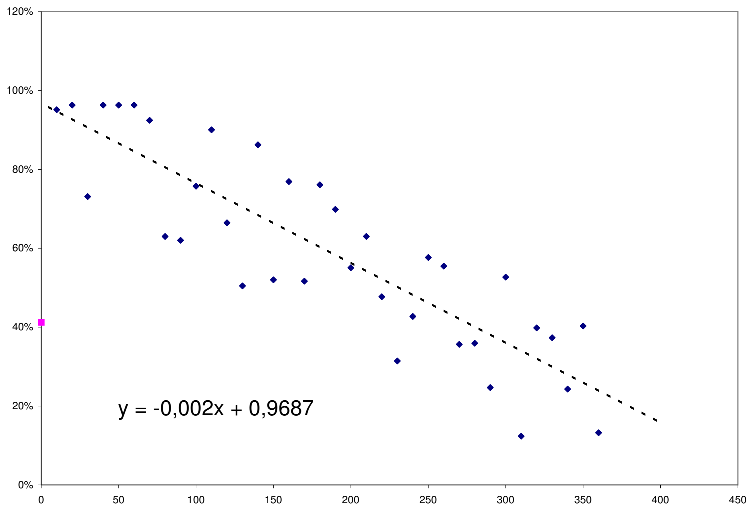


Figure 8.14: Scatterplot of complete population test of prediction accuracy with added trend line

trend line shows the accuracy of a single prediction is approximately linearly dependent on its Δ_c according to:

$$\text{Accuracy} = -0.002\Delta_c + 97,$$

where Accuracy is in percentages. The error associated with this trend line is 8 percentage points on average, resulting in an approximate 90% reliability of reliability prediction.

As a conclusion we can state that it is reasonably possible to associate a reliability with a single criminal career prediction using its curvature as the main indicator.

8.5 Conclusion and Future Directions

In this chapter we demonstrated the applicability of temporal extrapolation for the extrapolation of criminal careers. This method assumes that the visualization of a clustering inherently contains a certain truth value that can yield powerful results but reduces time complexity. We superimposed an extrapolation on an existing visualization of criminal careers and performed a series of tests that determined the speed and accuracy of the approach as well as the values of the necessary parameters. A clustering reduction was also performed to speed up calculation of a criminal career prediction.

It turns out that a clustering size of 10,000 criminal careers from the database can serve as a solid basis for extrapolation. If this extrapolation is accomplished by a spline extrapolation with straight line continuation and 50 reference points, accuracy can reach 88%. We also implemented and tested an approach that determines if an individual career is predicted reliably.

As time constraints are essential to successful implementation within actual police software services, it was important to reach significant gains in computational complexity. As an end-result, all necessary tasks to be repeated for every offender to be analyzed, can be completed in approximately 1 second.

Next to the fact that predictions can immediately be visualized to police end users because of their visual nature, offender's careers can be predicted with very high accuracy in a single second. These properties make the method very well suited for incorporation in background processes at police stations, allowing alerts to be sent to dedicated machines.

A weakness to the approach, that is native to extrapolation problems, is that the lack of enough information can cause very unreliable predictions, resulting in a minimum of 3 time frames of activity before anything valuable can come out of the extrapolation process. Unfortunately, data in the Netherlands is collected on a year-by-year basis, effectively establishing the demand that only third or higher year offenders can have their careers predicted.

The judicial constraints and privacy issues concerning the material in this chapter are discussed in Appendix A.

Future research will aim at reaching even higher accuracy values by improving the selection of reference items close to the extrapolation line. A triangular shape can, among others, be employed, that selects more reference points further away (more description of the future), or closer (more reliability) from the last known data point. Also, a search for common subcareers could be performed, that could reveal subcareers that "define" certain classes. These subcareers, especially if they occur in the beginning of a career, could possibly improve both the speed and accuracy of future career prediction.

Chapter 9

A Distance Measure for Determining Similarity between Criminal Investigations

In comparing individual criminal investigations on similarity, we seize one of the opportunities of the information surplus to determine what crimes may or may not have been committed by the same group of individuals.

For this purpose we introduce a new distance measure that is specifically suited to the comparison between criminal investigations that differ largely in terms of available intelligence. It employs an adaptation of the probability density function of the normal distribution to constitute this distance between all possible couples of investigations.

We embed this distance measure in a four-step paradigm that extracts entities from a collection of documents and use it to transform a high-dimensional vector table into input for a police operable tool. The eventual report is a two-dimensional representation of the distances between the various investigations and will assist the police force on the job to get a clearer picture of the current situation.

9.1 Introduction

In contrast to all applications discussed so far, that dealt with a single, rather static database stored at police headquarters, data mining is also particularly suited for usage on case related data, that is gathered for a single investigative purpose. Since tools that deal with this kind of data are confronted with data of an beforehand unknown structure or quantity, an entire new set of challenges need to be addressed for such innovations to be successful.

One of the key problems in this area of policing is the sheer amount of data that is stored on computers, that are nowadays confiscated more regularly during criminal investigations. A contemporary police force should be capable of dealing with this stockpile of data during the course of the investigation, yielding any valuable piece of information both timely and accurately. Usually, people working on these investigations have no technical background, so information should be presented to them in a comprehensive and intuitive way.

This chapter discusses new tools that deal with the extraction of logical concepts from police narrative reports and documents found on crime scenes in order to automatically establish an educated guess on what crimes may be committed by the same (group of) criminals. To this end we employ *text mining*, a *distance measure* and an *associative array clustering technique*. We discuss the difficulties in case-comparison and the specific distance measure we designed to cope with this kind of information.

The main contribution of this chapter is the discussion in Section 9.6, where the distance measure is introduced.

9.2 Project Layout

As discussed earlier, useful information exists in unstructured data like police narrative reports, intercepted emails and documents found on crime scenes. It would be desirable to employ this information for case comparison in order to supplement the forensic work done on-site and provide police officers with information about crimes that may be committed by the same perpetrators. However, this information is usually deeply hidden in the unstructured setup of such, often free-text, documents. Even after the employment of a suitable text miner, the enormous amount of extracted entities still poses many problems. As is common with police work, some cases suffer from a lack of data, while generate stockpiles of paper work. Comparing cases that differ extremely in size in terms of entities extracted from this unstructured data, is one of the challenges. Another is that of preparing the resulting data of this comparison, visualizing it and presenting it to the officers on the case. Our research aims to address both these challenges and to set up a police framework for comparing cases on the basis of (collected and created) documents.

9.3 System Architecture

Our "case-comparison" system is a multiphase process that relies on a commercial text miner, a table transformation unit, a distance calculator and a visualization tool. We therefore describe our process as a *four-step paradigm* (see Figure 9.1) and will elaborate on the individual components and their in- and output in the following sections. Black boxed, the paradigm reads in a collection of unstructured documents and provides a comparison report to the end user.

The documents we use as input for our case comparison system consist of two different types, both of which are provided by the individual regional police departments for analysis:

- Police narrative reports: one of the types contained in our document collection is that of the police written narrative reports. These reports are created to describe

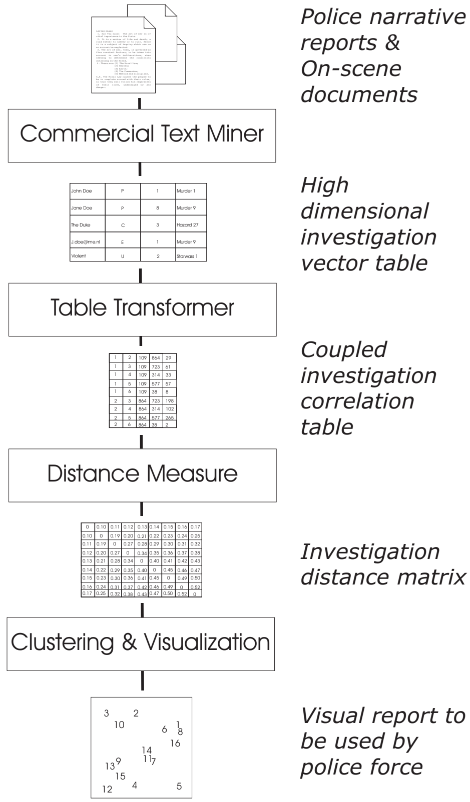


Figure 9.1: Four-step paradigm

a crime, the people involved and the Modus Operandi (MO). Protocols exist how these reports should be written, but these rules are not always strictly followed. Also, these reports suffer from an abundance in police terminology (for example, almost all reports contain the words “rep.” (report) and “serial number”) and they are likely to have typing mistakes in for example the way names are written. Some of these spelling mistakes are intentionally introduced by suspects to avoid cross referencing. Due to these effects, the police narrative reports are often reasonably polluted.

- Crime scene documents: digital documents found on crime scenes are often very rich in information. They contain valuable information like email contact lists that can give an idea of other people involved or lists of goods acquired to commit crimes. Since they are mostly created by the perpetrators themselves they are less likely to have errors or typing mistakes. Therefore, the crime scene documents are less noisy than the narrative reports, but are unfortunately also more rare.

Table 9.1: Different types recognized by entity extractor

Type	Description	Percentage
K	License plate	0.90%
P	Person	5.34%
u	URL	0.02%
O	Organization	0.48%
L	Location	0.69%
e	Email address	0.04%
D	Product	0.03%
i	IP address	0.02%
U	Unknown	92.45%

When processing these documents we first subdue them to a text miner that yields a table with concepts, the number of appearances and the investigation they belong to. This table is then transformed to a high-dimensional vector space, where each vector represents a different investigation. We then extract comparison numbers in our transformation unit, that is presented to our distance calculator. The distance matrix that results from this is now fed to the visualization and presentation tool, that can be operated by the analyst himself. The results will be presented visually and are ready to be interpreted and used by the individual police department that provided the documents.

9.4 Entity Extraction and Table Transformation

An important step in the process of getting from a collection of documents to a comparison overview is that of entity extraction or text mining. As mentioned earlier some specialized tools were created by different research programs. The INFO-NS program [28] suggests a framework for evaluation of commercial text mining tools for police usage. In practice, a lot of police departments employ one of these commercial suites for their data mining endeavors. In order to comply with this situation, we chose to employ the use of the SPSS Lexiquest text mining tool [41] as the starting point for our comparison framework. Through a simple operating system script the documents are fed into the text miner one investigation at a time. The text miner then yields the following table:

<u>Entity</u>	<u>Investigation</u>	Type	Amount

In this table Type refers to one of the types defined in Table 9.1 that also shows the percentage of these types in the dataset used for our experiments. The resulting table is primary keyed by both entity and investigation but since the final objective is the comparison of investigations it is necessary to transform this table to an investigation based one that contains all characteristics per investigation. The table should therefore contain information about what entities are present in each investigation. The table we are creating therefore has an integer field for every entity. If the key investigation-entity is present in the original table, the corresponding field in the new table will be equal to the contents of the amount field and 0 otherwise. The number of distinct investigations we can retrieve from the table will be denoted by m. This yields the following high-dimensional vector table:

Investigation	Entity 1	Entity 2	...

where the number of dimensions, apart from the key attribute Investigation, is equal to the number of distinct entities in the previous table, which we will denote by n.

This table contains highly accurate entity usage information per investigation, for it contains all available information. We can now employ this information to get insights into the way different cases are alike in terms of used concepts.

9.5 Multi-Dimensional Transformation

The high-dimensional table that resulted from the previous step can now be used to describe distances between the various investigations in n-dimensional space. Naturally, we need to transform this table into a two-dimensional representation of the given data in order to come to a useful visualization in our tool. In order to achieve this dimensional downscaling we assume similarity can be constituted by the sizes of the investigations in terms of entities extracted and the entities they have in common.

According to this assumption, we compare the investigations couple-wise on each individual entity (see Figure 9.2) and score the couples on the amount of common entities according to the following method: every time both investigations have a value larger than or equal to 1 in a column, the score for overlapping is raised by 1.

The algorithm treats every investigation as a subset of the total set of entities. The calculation of the amount of common entities in two investigations is therefore synchronous to the calculation of the amount of items in the intersection of both sets (see Figure 9.3), and goes as follows:

$$\text{Overlap} = |\text{Inv}_1 \cap \text{Inv}_2| \ ,$$

where Inv_i is the set of entities for investigation i ($i = 1, 2$). We also let Size_i denote $|\text{Inv}_i|$.

It is possible to utilize a filtering technique at this point to exclude all entities with type “U” (unknown) from the comparing method. This will probably yield highly accurate results, due to the high expressive power of the recognized entities. For example, two cases sharing the person “John Doe” are more likely to be similar than two cases sharing the word ‘money’. However, due to the high percentage of entities that are being classified as unknown, leaving them out can cause undesired shortcomings to the algorithm.

	John Doe	Jane Doe	J.doe@me.nl	The Duke	555-123456	Violent	Consigliere	1 Mystreet	Money	Narcotic	...
Case A	0	0	0	0	0	0	2	0	0	0	...
Case B	0	1	0	0	0	0	2	0	0	0	...
Case F	1	1	3	1	2	0	0	0	0	0	...
Case G	0	0	0	2	0	0	0	1	0	0	...
Case I	2	0	0	0	1	0	0	0	0	0	...
Case K	0	0	2	0	0	1	0	0	0	0	...
Case M	0	0	1	0	0	0	3	0	0	0	...
Case N	0	0	0	0	0	0	0	2	0	2	...
Case Q	1	0	0	2	0	4	0	1	0	0	...

Figure 9.2: Comparing the investigations on common entities

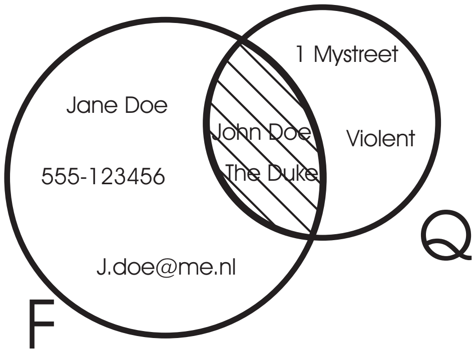


Figure 9.3: Viewing the transformation process as calculating intersections

For example: the word 'violent' may well be a keyword in comparing two individual investigations, but is still categorized under type 'U'.

The mentioned algorithm provides a table with $\frac{1}{2}m(m-1)$ rows, where each row represents one couple of investigations. Each row consists of three columns: the size of the first investigation, the size of the second and the amount of common entities:

$Size_1$	$Size_2$	Overlap

This table is comparable to the ones used in data mining on shopping baskets where the goal is to determine which customers exhibit similar shopping behavior.

9.6 Distance Measure

To constitute similarity between different investigations we introduce a distance measure, that calculates the distance between two such criminal cases. The distance measure we propose is a function over the parameters we stored in the table resulting from the previous step. The higher the function outcome and thus the distance, the less alike two investigations are. Our function yields a distance value between 0 and 1.

It is not just the amount of common entities that constitutes the distance between two investigations; the different sizes of the investigations should be taken into account as well. It is common practice in for example the analysis of the earlier mentioned shopping baskets, to let a difference in size have a negative effect on similarity. If we take a look at two shopping baskets, and we observe that one basket contains a newspaper and a bottle of wine and another basket contains the same paper and wine but also a hundred other items, no analyst would claim similar shopping behavior of the two customers, although 100% of one of the customer's acquisitions is also in the other one's basket. Therefore, distance measures like the symmetrical distance measure [26]:

$$\frac{(\mathrm{Size}_1 - \mathrm{Overlap}) + (\mathrm{Size}_2 - \mathrm{Overlap})}{\mathrm{Size}_1 + \mathrm{Size}_2 + 1}$$

that also incorporates size differences, are often employed in this area. However, this does not hold for the comparison of investigations. Although the size in terms of entities extracted may well be an indication of difference between two cases (many or few computers found on scene) it is, as mentioned earlier, not uncommon for law enforcement cases to differ largely in size while they still involve the same people (the police was at the scene very quickly vs. the criminals had time to destroy evidence).

As a consequence, the symmetrical distance measure mentioned above is not applicable in the area of case comparison. Naturally, the sole use of common entities is not applicable either. We therefore introduce a new distance measure specifically suited for the comparison of criminal investigations based upon entities extracted.

We propose a distance measure based upon the random amount of common entities two investigations would have if they were drawn randomly from the entire set of entities. The deviation between the size of the randomly selected intersection and the actual amount of common entities then constitutes distance. The size of the entire collection of entities, the universe of entities, will be denoted by A. In calculating this value we only count each distinct entity once instead of using each occurrence of a single entity in the table. This will more accurately represent the probability space for each individual investigation subset. We will denote the average size of the intersection of two randomly drawn subsets having sizes X and Y as E, which can be calculated as follows:

$$\frac{X}{A} \cdot \frac{Y}{A} = \frac{E}{A} \quad \Longleftrightarrow \quad E = \frac{X \cdot Y}{A^2} \cdot A = \frac{X \cdot Y}{A} \quad .$$

We can now easily calculate the difference (Differ) between the actual value Z and the expected value E as follows:

$$\mathrm{Differ}(Z) = Z - E \quad .$$

As is clear from the calculation of E, the expected value depends on the three variables X, Y and A. As a consequence, a very large universe A can lead to very low values of E and thus to very large differences between E and Z. This variation can be considered to be disruptive to the process in the following two cases:

- Some very large investigations without any relation to the other investigations, for example, two large Finnish investigations among a series of English investigations, are included in the list to be analyzed. The large number of unique entities these Finnish investigations would contribute to the universe A would implicate that all other investigations would have very low expected values and therefore very high differences. This can put all those investigations at a distance from each other that is far less than it should intrinsically be.

- When a lot of different investigations are to be compared, they all contribute a number of unique entities to the universe A. This means that, while the actual chance of two investigations having a certain overlap does not change, the calculated E would decrease approximately linearly to the amount of investigations included in the analysis. This can, as was mentioned above, lead to too small distances between the investigations.

As a measure for countering these effects we propose to calculate A from the actual overlapping values instead of just using the total amount of entities. We have implemented this method and compared it to the standard method described above.

We will base the alternative calculation of A upon the actual overlapping entities in all the possible couples, meaning that we calculate A out of X, Y and Z, instead of calculating E out of X, Y and A. Our method will then average all the individually calculated A's and use this number for A instead of the total amount of entities. Calculating A will go as follows:

$$A = \frac{\displaystyle\sum_{i=1}^{m} \sum_{j=i+1}^{m} \frac{X_i \cdot Y_j}{Z_{ij}}}{\frac{1}{2}m(m-1)} \tag{9.1}$$

In this summation we omit the pairs (i, j) with $Z_{ij} = 0$. Having obtained the differences we can calculate the distance between two investigations.

The normal distribution is the most widely used distribution in statistics and many statistical tests are based on the assumption of normality. One of the most used representations of this distribution is the probability density function (see Figure 9.4), which shows how likely each value of the random variable x is:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right),$$

where σ denotes the standard variation and μ denotes the mean. Since we want to give a quantization of how notable the Differ function outcome is, we can take this function

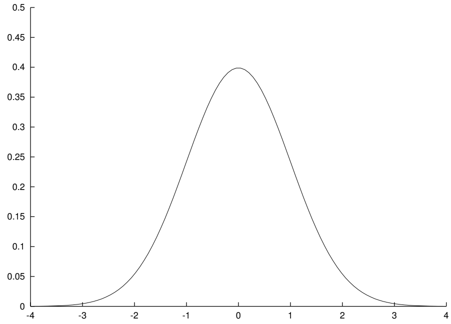


Figure 9.4: Normal probability density function; $\sigma = 1, \mu = 0$

as basis for our distance function. We will thus employ an adaptation of that function to calculate the distance of two investigations by using the above mentioned function Differ. First, because we want to normalize our outcome between 0 and 1 we need to top of our function at $\frac{1}{2}$ by changing the factor before the exponential part of the function into $\frac{1}{2}$. Then we take the minimal value of X and Y as our standard deviation, since it is logical for the intersection of two large subsets to deviate more from the expected value than the intersection of two smaller subsets. We can flip the function part left of the mean to represent a positive deviation as a smaller distance between the two investigations. If we denote $\min(X, Y)$ as the minimum value of X and Y, our final distance function will look like this:

$$\mathrm{Dist}(Z) = \begin{cases} \frac{1}{2}\exp\left(\frac{-\mathrm{Differ}(Z)^2}{\frac{1}{2}\min(X,Y)}\right) & \text{if Differ}(Z) \geq 0 \\ 1 - \frac{1}{2}\exp\left(\frac{-\mathrm{Differ}(Z)^2}{\frac{1}{2}\min(X,Y)}\right) & \text{otherwise} \end{cases}$$

This function calculates distance with respect to any size difference that may occur between two investigations while not incorporating any negatives effect for that difference. The proposed measure is symmetrical (X and Y can be exchanged). If two investigations are very much alike, their distance will approximately be 0; if they are very different their distance approaches 1.

As is clearly illustrated in Figure 9.5, the form of the graph of the distance function differs significantly between different sized investigations. This enables us to compare different sized subsets, since similarity is constituted by the probability space rather than integer values. For example, three overlapping entities in two small subsets can now judge the two cases to be just as similar as 10 overlapping entities in a large and a small subset, or 100 overlaps in two very large cases.

If we apply this distance measure to all the rows in the last table we are able to create a distance matrix M, where for each $1 \leq i \leq M$ and $1 \leq j \leq M$ element M_{ij} represents the distance between investigations i and j. Due to the fact that the distance between i and j is the same as between j and i our distance matrix is symmetrical. Having calculated all the distances we can display the investigations in our visualization tool.

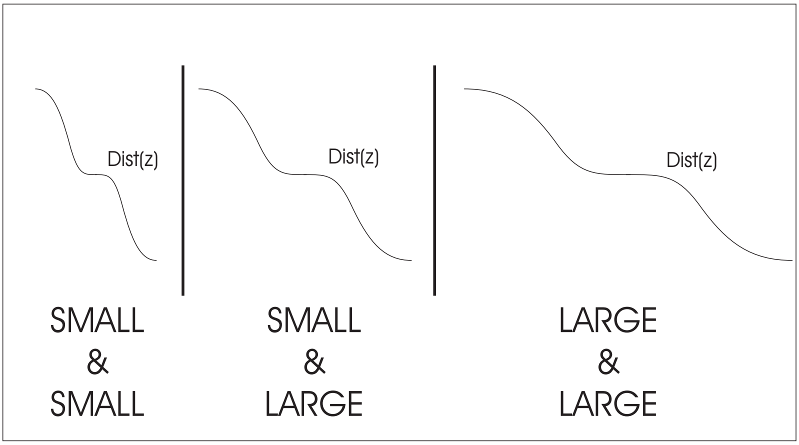


Figure 9.5: Distance function of different sized couples of investigations

9.7 Visualization

It is desirable for police personnel to be able to view the total picture in one glance. In most cases it is not possible to display a high-dimensional situation, such as our initial vector table, perfectly in a two-dimensional plane, especially after the amount of transformations our data went through. We therefore employed the associative array clustering technique [27] to display (an approximation) of all the distances between the different investigations in one two-dimensional image. This technique can be viewed as Multi-Dimensional Scaling (MDS, see [15]), and is especially suited for larger arrays. The image we now have can be fed back to the officers on the case to enhance their understanding of the situation.

The associative array visualization technique strives for the creation of a flat image of all considered elements where the physical distance between them is linearly related to the distance in the matrix, while minimizing the error in that distance, sometimes referred to as “stress”. It is an iterative process that starts off at a random situation and through a specified number of iterations tries to improve that situation until it reaches a more or less stable state. This is a state where the error made in the placement of the elements is at a (local) minimum.

The algorithm works as follows: starting at the earlier mentioned random position, where all the elements are in an arbitrary position, the algorithm investigates a random couple of elements and when the distance in the image is relatively larger than the requested distance, the pull operation is executed. If, on the contrary the current distance is smaller than the distance in the matrix the push operation will push the elements away from each other. As can be seen in Figure 9.6 the push and pull operations move the elements in the target image away from or towards each other on the line defined by the two elements.

In every iteration all couples of investigations will be evaluated and their respective

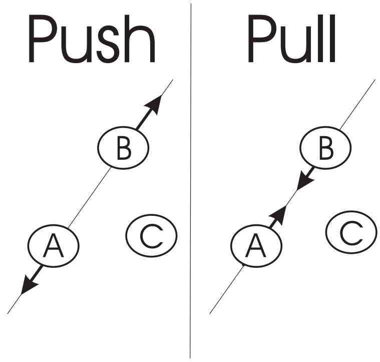


Figure 9.6: Push and pull operations

distances corrected. Since the image usually can not be displayed entirely correct in the two-dimensional plane, the image might differ a bit depending on the random starting point, but is consistent enough to give a good overview of the similarity between investigations. Also note that rotations and reflections may be applied without affecting the outcome.

It is imperative for the tool that performs this task to be operable by the police officers that request the similarity analysis. The tool to be designed therefore needs to be extended by a graphical user interface (GUI). Developing a GUI not only serves the purpose of making the application usable by police personnel, but also gives insights in the formation of the image and enables us to detect problematic situations and improve the algorithm.

The tool we developed allows a number of settings and has a main screen where the user can see the image unfold in different speeds. The user can then output the images in PDF format for usage in a document. The user can customize the screen to display investigation labels of numbers if the titles overlap too much.

As a simple demonstration of the algorithm's possibilities, we tried to regain the image of four points forming the corners of a square and a fifth point in the center, by means of its distance matrix. The result depends on the random starting position of the five points and if done correctly would represent the original image reasonably accurately in compliance with rotation and mirror symmetry. Figure 9.7 is one of the possible results.

9.8 Experimental Results

One of the major tasks of the police is the dismantlement of synthetical drugs laboratories. Several of these have recently been located and rendered out of order. Investigation of these crime scenes has led to the acquirement of digital documents, interception of email traffic and the compiling of numerous narrative reports by officers and forensic experts assigned to these investigations. Given the nature of the different laboratory sites, case detectives suspect common criminals to be involved in exploiting some of these

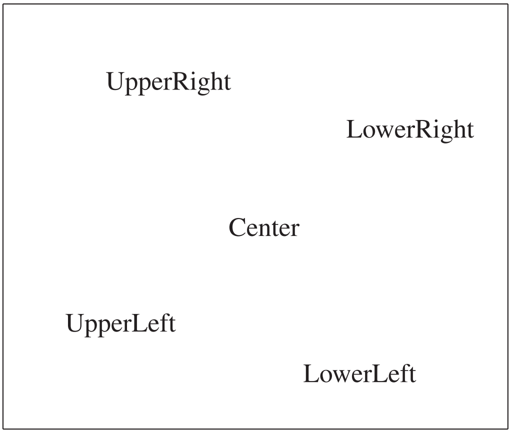


Figure 9.7: Visualization of a centerpointed square based upon its distance matrix; visualized using labels

locations for the creation of synthetical drugs. Employment of a clustering technique should provide answers to the questions about common perpetrators in these and future cases. Research on the collected data should therefore focus on the following:

- Producing a comprehensive report on the similarity of current investigations into the construction and employment of synthetical drugs laboratories.

- Using the data to produce a tool that enables the police to perform similar tasks in similar future situations.

As was mentioned earlier, the data in such documents are often polluted by the inclusion of enormous amounts of police terminology and the large number of typing mistakes. Also, since the commercial text miner is not specifically trained on police documents, a lot of entities were labeled as unknown or as the wrong type. Incorporating this knowledge into our scripts we decided to use all types of entities instead of the inherently more powerful classified entities alone. We present some results for this analytical task containing $n = 28$ police investigations, together having $m = 152{,}820$ distinct entities. Here, A is either m, using just the amount of entities, or is computed according to Formula (9.1).

Usage of our new distance measure on this data yielded distance matrices that indeed showed results that could indicate similarity between some of the individual investigations. The distance matrices showed some significant difference in distance values between the individual investigations. Application of our clustering approach to this newly generated matrix for both different calculation methods for A showed a clustering image (Figure 9.8 left and right) that indeed demonstrated that certain investigations are closer and therefore more similar to each other than others. We infer from these images that there is some relevant similarity between certain investigations and submitted the reports, outputted by our application to the investigation teams. We are currently in discussion with the domain experts about the validity of the outcome of both methods employed by our system.

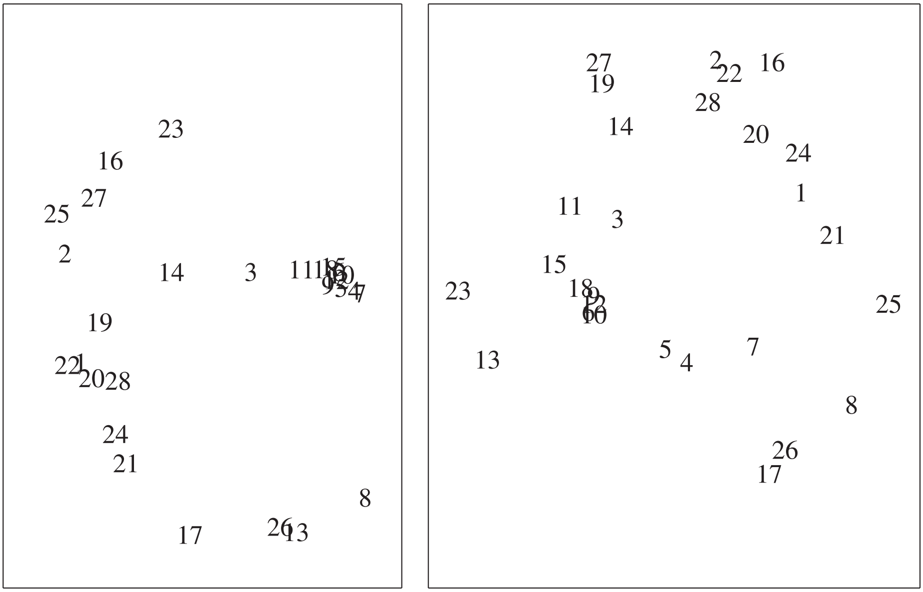


Figure 9.8: Visualization of the database of investigations using $A = n$, numbered 1 to 28 (left) and visualization of the database of investigations using Formula (9.1) for estimating A, numbered 1 to 28 (right)

In the comparison of both possible methods of calculating A, it is noteworthy that the distances represented in both images do not vary a lot between the different methods but show minor differences, as is, for example, evident in the placement of 13 in Figure 9.8.

9.9 Conclusion and Future Directions

Data mining is a suitable solution for many problems and opportunities arising from the information explosion. In this chapter we demonstrated the applicability of data mining in the comparison of individual criminal investigations to establish a quantity for similarity between them. We used a four-step paradigm to transform a set of documents into a clustering image that gives a full overview of the similarity between all investigations and is ready to be used by police experts. The new distance measure we introduced was specifically designed for this purpose. It incorporates the differences in information size between investigations while still maintaining a realistic comparison standard.

Future research will aim at getting a clearer picture about the computation method for A. Assigning n to A describes the situation with a true to reality universe of entities while using Formula (9.1) probably delivers better end-results for largely different or a large number of investigations. Both methods of assigning a value to A therefore have their own merits and more testing on different data sets is a prerequisite in deciding between them.

The commercial text miner used in this project was a source of problematic entries in our initial table. Incorporation of domain specific text miners such as used in the COPLINK project [12] would probably lead to a significant improvement of the total system. This is one of the subjects where future research should focus on.

Extending our research in this area by creating a linguistic model that deals with the large amount of typing mistakes would be a great advantage to our entire approach and would probably lead to an even more realistic end-report and a clearer picture that the police force has of the perpetrators it pursues.

Chapter 10

Identifying Discriminating Age Groups for Online Predator Detection

Children are a major user group of today's internet, mostly focusing on its social applications like social networking sites. Their online activities are not entirely without risk, as is demonstrated by the presence of online predators, who misuse its potential to search for and contact them with purposes in the area of sexual abuse. In this chapter we investigate the presence of these predators on *Social Networking Sites* and present a method of automatically detecting them. In this process we adopt a genetic algorithm approach that discovers a relation between the amount of friends on an online profile and the age of its owner, searching for statistically significant differences between known offenders and a control group of standard users. After the creation of this model, thresholds are calculated that classify users into danger categories. Experiments on actual police data sources show some promising results in countering this stealthy threat.

10.1 Introduction

The growth of the internet and the incorporation of its usage in daily life has been especially strong in the teenage demographic. Children in this age group are young enough to view the internet as an integral part of society, yet old enough to fully utilize its potential, especially in the social area. The onset of Web 2.0, the change of the World Wide Web from a static archive of websites into a dynamic environment with interactive programs and websites, has greatly contributed to the way contemporary youth structure their social life. An illustrative example of such a concept would be that of Social Networking Sites (SNS); user profile based dynamic websites that can be used to set-up, maintain and manage networks of (online) friends.

Although the benefits of early accustomization to technology are plenty, internet usage by young people is not entirely without risks, especially since most of the activities are performed alone and usually without adult supervision. Recently there has been a raise in awareness of such dangers and numerous (governmental) campaigns are underway to inform teenagers and their parents of a number of internet hazards, of which the *online predator* is the most prominent. Seen from a general point of view, these predators, “hunt” the internet for other users to exploit, acquiring some asset from them, which can be anything, ranging from monetary gains to identity theft, but most of the time, the term is used to describe the more specific type of *online sexual predator*. These are people that search the internet for sexual prey, mostly children, with the eventual goal of committing some kind of sexual offense. The SNS mentioned are very well suited for such activities, providing both anonymity and a large, easily searchable and publicly available database of possible victims that are also easily contacted, using the website’s standard functionality.

Naturally, this is an area of criminal activity that is currently monitored by police (internet-)detectives. Unfortunately, methods employed by these divisions are mostly passive, dealing with observation of previously encountered online offenders or in reaction to civilian narrative reports on a certain case. It could be a great asset to device a method that can more actively seek out potential offenders through monitoring systems that detect certain behavioral patterns in an online environment, such as SNS, and classify users of that environment in potential danger categories. This chapter aims to provide an initial step toward achieving systems that deal with this issue. It describes the potential ability to detect predators based upon an individual’s age and the percentage of under aged “friends” this person has on his or her SNS profile. Some of the issues are discussed and a tool for accomplishing this task was tested on actual police data to determine its suitability for and precision in the detection of online predators on SNS.

10.2 Background

In order to gain a good understanding about the way online sexual predators employ the possibilities of SNS it is imperative to investigate both the (behavioral) properties of these offenders and the main idea behind these interactive websites, providing as much information as needed about the functionality that can be (mis)used within these environments.

10.2.1 Social Networking Sites

Social Networking sites are defined as follows [8]: SNS are web-based services that allow individuals to

1. construct a public or semi-public profile within a bounded system,
2. articulate a list of other users with whom they share a connection and
3. view and traverse their list of connections and those made by others within the system.

Thus, being a typical example of Web 2.0 [33], users are the main source of content within SNS, being responsible for its creation and maintenance, the service provider being only responsible for the development and maintenance of the framework. In practice this means that people can register at the SNS by using a nickname (username, profilename) after which they can build a personal page, sometimes with a picture of him- or herself. Physically, this page is located on a server of the SNS and can be setup and changed by the user through a *content management system*, which eliminates the need for knowledge of web languages like HTML. Usually, information like hobbies, geographical location, work and personal situation is added to the page which enables other SNS users or, in the case of a public profile, all internet users to check if the newly registered individual has common interests or lives in the same country or area. It is usual for a SNS to provide the possibility to limit the access to a certain profile to SNS members or even to a group of specified people

As stated above, the core of a SNS is the possibility to create links between one profile and a (large) number of other profiles to state that a certain connection exists between the owners of both profiles. These connections can vary from real-life friends, online friends to club memberships and corporate employeeships. The nature and nomenclature of these connections may vary from site to site, but in this chapter we will refer to these connections as *friends* of the profile holder. Note, that is it is often not possible to link to profiles on different SNS, effectively limiting users to have friends within the same framework alone. The list of friends is usually publicly viewable so profile visitors can immediately see who the (other) friends of that specific person are. It is not always the goal of a SNS user to meet new people. A lot of SNS users only use the SNS to communicate with people they already know and also have contact with in real life. There are also SNS users who add almost anybody to their friends list with a result of having hundreds or even thousands of friends.

Most SNS provide a search option that allows users to search for people they know, for example a search by name, or for profile holders that adhere to a specific set of properties specified by the searching party. Sometimes, the friend list of other users can be searched in this manner as well. Within the scope of this research it is noteworthy that a search on age option is common, providing an easy way to specify certain “preferences” a predator might have when searching for potential victims.

10.2.2 Online Identities

One of the problems in today’s internet that certainly holds within SNS is that of “identity”. In real life, identity is defined as [3]: “The distinguishing character or personality of an individual.”

This definition cannot be translated directly to an online environment, where it is possible to maintain a profile that resembles your offline identity most closely or to craft an entirely new profile that suits a specific purpose, for example when applying for jobs, meeting people of the other gender or, in the case of the online predator, to mask one’s true identity to reach out more easily to potential victims. When communicating in the physical world the human body is automatically connected to identity. When talking to one another, people can therefore easily confirm that the person who they are communicating with, is really who (s)he says (s)he is [18]. This is one of the reasons why communication on the internet and especially on SNS, where physical recognition is limited to visual media chosen by the user under observation him- or herself, can be easily used to mislead people. Also, the inability of detectives to successfully match all known offenders to online profiles poses a significant surveillance issue. Therefore, the difference between the physical identity and the online identity, plays a crucial role in criminal cases related to SNS.

The difference between offline and online identities can have advantages as well. Law enforcement agencies around the world can use covert identities, feigning for example that they are a 12 year old boy that wants to meet the individual under observation in a hotel. On arrival the offender can then be arrested by law enforcement officials. Especially in the United States methods like these have been successfully employed [31], but they are often outlawed in other countries due to entrapment issues. In these countries a more passive monitoring approach with early warning systems in place would probably yield the best results in countering the threat of online predators. A good starting point would be to analyze the demographics of these offenders.

10.2.3 Online Sexual Predator

Comparable to the animal world, a predator in the human world “hunts” other individual for some commodity, in the process hurting the other person’s interests, identity or even life. The scope of this research is limited to the area of sexual predatation, where the modus operandi is set in an online environment. In literature, both the term online sexual predator and online predator is used to describe the same phenomenon. Although a small percentage of victims concerns adults, the online predator mostly preys on minors or under aged individuals, which is the type of predatation this chapter focuses on.

Child sexual predators are a heterogeneous group, and as a result it is difficult to define a typology of the sexual predator [17, 16]. Psychologists and law enforcement investigators employ the following, somewhat crude definition [20], which claims a typical sexual predator is likely to be:

- a male,
- aged 18–72 (preponderance between 30 and 42),
- having a successful career (middle to upper management),
- being a college graduate,
- being married or divorced,
- being a parent himself (but whose children are older than his targets).

It is plausible that online sexual predators are younger of age compared to the traditional sexual predators mainly because of the usage of modern technology. Research done in this area tends to support this [38, 22]. When looking at the above mentioned figures, we can assume the typical online sexual predator is a male, aged between 18 and 65.

In contrast to the standard *grooming* approach, the process that bridges the gap between being strangers and having intimate physical contact, online sexual predators mostly use non-traditional ways to get in contact with children, searching the internet on chat rooms and websites specifically meant for minors. While searching, the predator tries to find individuals who fit the age, sex and location of their preference. When a potential victim is found, contact is usually made through chat [30]. A likely next step in the online grooming process is a potential inclusion in the predator's network on a SNS.

10.3 Predator Presence on Social Networking Sites

To analyze activities of online predators on SNS it is best to focus at a specific SNS. A lot of the SNS basically work similar, providing the same structure and functionality, therefore results yielded by the following research are assumed to be valid for most SNS. In this research we chose to investigate the presence of online predators on the largest and most used SNS in the Netherlands: Hyves (`www.hyves.nl`). This SNS is used by people of all ages with a strong group of people under 35.

To get an impression of the amount of online predators being active at Hyves a list of known offenders was extracted from the national database of the Child Pornography Team of the Dutch National Police Agency. To make the list manageable, only offenders were added who are known for being active on the internet and fall within the demographics described above; consequently the offenders in this group can be called potential online predators. This shortlist served as a basis upon which a brute force search was performed by hand to match these individuals with a high degree of certainty to a Hyves-profile, comparing the data on first name, last name, home town, age and date of birth. All possible matches with a slight degree of doubt were omitted from the result list. Obviously, people with a fictional profile name, address, age, etc were either not identifiable or excluded due to doubt, so the quantifications displayed in Figure 10.1 can be viewed as a lower bound of predator activity levels on this SNS.

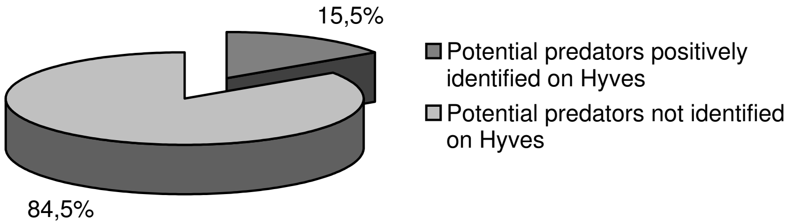


Figure 10.1: Predator Activity on Hyves based upon shortlist

Figure 10.1 suggests that at least 15% (316) of 2,044 child sexual predators on the shortlist are active on this specific SNS alone. This situation poses a significant risk for children using the same site as well. This is confirmed by the increasing number of narrative reports and reports filed at different hotlines (see Figure 10.2).

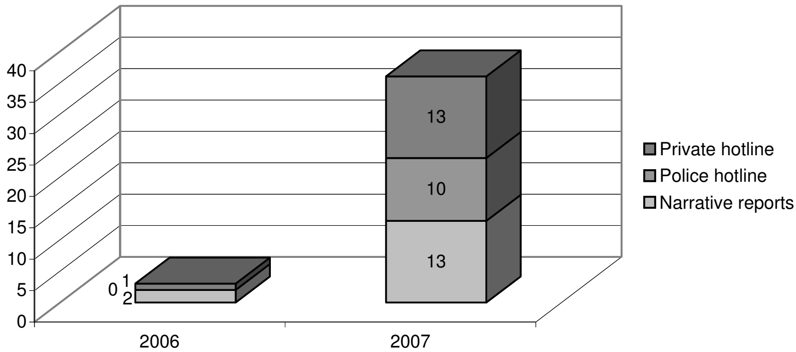


Figure 10.2: Increase of reports on predator activity on Hyves

A number of reports indicated that a parent ran into a profile of an unknown adult on his or her child's *friend list*, noticing that this person's profile contained a lot of other minors as well. Although, logically, no legal action can be taken against such individuals, this phenomenon gives rise to the assumption that a certain percentage of under aged friends can be a good indication of an individual's chance of belonging a certain danger category.

10.3.1 Amount of Under-Aged Friends: A First Glance

As a first step in the automated discovery of such a relationship, the total amount of friends and the number of minors on that list was retrieved for every offender on the shortlist. In addition, 960 random Hyves users were selected (10 individuals, of both genders, of every age between 18–65), who serve as a control group in this experiment. The same information was retrieved for this group, which resulted in Figure 10.3.

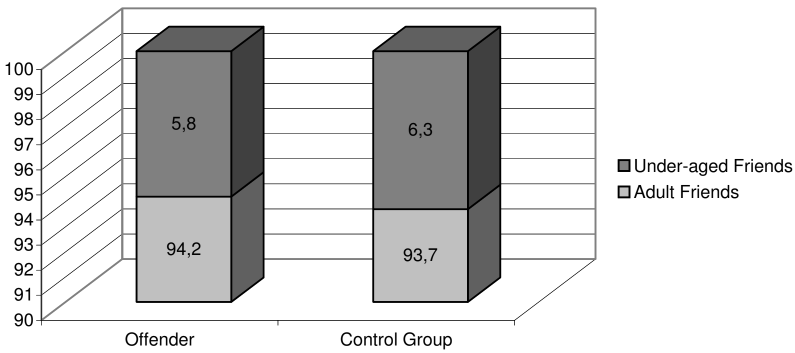


Figure 10.3: Average percentage of under-aged friends

At a first glance, the data represented in this figure rejects the current hypothesis, showing no significant difference and even a slightly larger percentage of under-aged friends for the control group. However, a more thorough investigation of the offender group reveals the reason for this perceived equality in variance: offenders in a lower aged sub-group (18–25) are underpresent compared to their control group counterparts, while the percentage in this group is relatively high compared to higher age groups. Hence, a more detailed analysis is warranted.

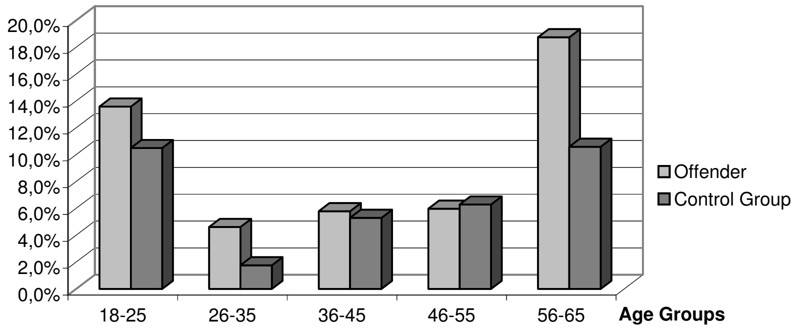


Figure 10.4: Average percentage of under-aged friends per age group

Figure 10.4 clearly shows that in the majority of the age groups the offender group has a larger percentage of under aged friends. However, not all the differences can be designated as significant. Furthermore, the arbitrary nature of the currently investigated age groups, makes it hard to draw any conclusions about the possibility of danger category classification based upon this property of a SNS profile. A method of automatically generating and evaluating age groups with a significant difference would greatly contribute to this cause. Moreover, the aggregation of threshold values for classification out of these discriminating groups has the potential to greatly improve police monitoring activities.

10.4 Approach

The percentage of under-aged friends varies greatly between the different age groups, both in the offender and the control group as is demonstrated in Figure 10.5. In this image, great differences can be observed between the different age groups, for example between predators of 25 and 35 of age. The bottom image zooms in on the bottom part of the graph to enable a clearer observation of the differences in the lower range of percentages. However, the graph clearly reveals a large schism between the percentages of the two different groups for some ages. An age-wise comparison is therefore necessary to examine a possible relation between the two. During this process, age groups can be constructed that clearly discriminate between the two groups, due to the significance of the difference in percentage in their sample populations.

There are two properties of the experiment that make the recognition of such groups non-trivial. First, not one of the ages supports samples large enough to validate a statistical test on significance for this single age. Hence, no conclusion can be drawn about

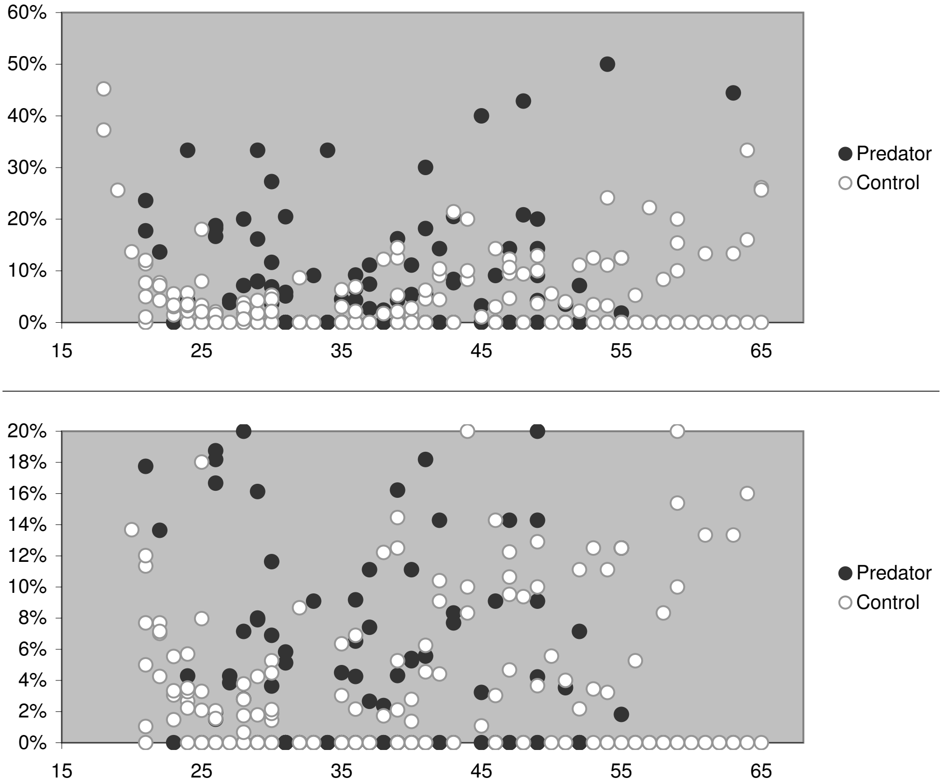


Figure 10.5: Scarce scatter plot for separate ages

the discriminative properties for a single age and grouping must occur on this variable before significance can be established with any certainty.

Second, the variance within groups for the age variable is large, which makes the grouping of single ages for the establishment of a discriminative property difficult. Furthermore, it is not always clear beforehand what effect this variance will have on the discriminative property of two groups (possibly of size 1) that are about to be merged for evaluation. An example of this can be seen in Figure 10.6, where both ages show a clear separation between the control and offender group, but merging might or might not lead to a group that retains this significance in difference because of “overlapping”. As can be observed clearly, a separation line can be drawn in both Age group 1 and age group 2, but when merged, such a line can no longer be drawn without to much instances appearing in the wrong group. This effect occurs every time two separation lines are on different percentages and a large number of instances has a percentage very close to these lines. A special search method for these combined age groups is therefore warranted.

For this purpose we propose a simple *genetic algorithm* [19] that attempts to identify age groups that are both as large as possible and satisfy the significant difference require-

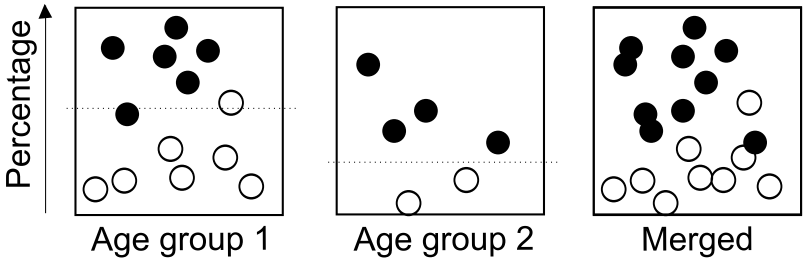


Figure 10.6: Percentages for both groups might overlap if merged

ment. After a certain amount of evolution cycles, the process yields a number of disjoint subgroups that satisfy the significance demand. They should contain the age groups that offer the best discrimination between the offender and control group and can be used for classification of new individuals. At the end of our approach (see Figure 10.7), the results will be tested against itself through a ten-fold cross validation method.

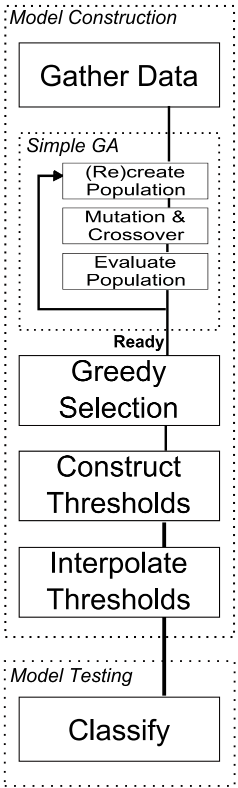


Figure 10.7: Approach

10.4.1 Genetic Algorithm

A genetic algorithm is a programming technique that mimics biological evolution as a problem solving strategy. Its foundations lie in a number of potential solutions (*candidates*) to that problem, and a metric called a *fitness function* that evaluates each candidate quantitatively, resulting in its *fitness*. Initial candidates are often generated randomly. After evaluation, all promising candidates are kept and allowed to reproduce. This is done based upon some pre-established operators, that most often include *crossovers*, combinations of existing candidates, and *mutations*, which are copies of existing candidates with slight modifications. Again, the most promising candidates are kept and the cycle continues in this manner for a large number of generations. The expectation is that the average fitness of the population will increase each generation, eventually resulting in excellent solutions to the problem under observation.

After the data selection, which was discussed in Section 10.3.1, candidate age groups were selected as starting population of the genetic algorithm. All possible combinations of two consecutive ages were considered candidates, resulting in a population of 46 candidates. An *elitist* selection method was adopted in order to preserve any older candidates with higher fitness than that of the newly generated candidates. Although this increases the risk of obtaining a *local optimum*, a situation where small steps backwards, needed for a potential large step forward, are immediately discarded, it speeds up the algorithm by preserving good, already realized, candidates that could potentially be lost by a series of unfortunate operator applications.

Operators

In every generation, the fittest candidates will be used to construct the new generation in three distinct ways, two of which are mutations and one that is a crossover.

The first mutation involves exchanging one particular age in the selected candidate group with a randomly selected other age, that is currently not in the group. Naturally, this will not affect the size of the group under observation. During this process, a copy of each of the before mentioned candidates is subjugated to this mutation resulting in 46 extra candidates.

The second mutation either removes a random age from a copied group or adds a random age to this group. Both options have an equal chance of being selected. Note that this mutation changes the size of the group, increasing or decreasing it by 1. This mutation also introduces 46 new candidates to the population. Note that no group is ever reduced to size 1 or lower.

The crossover operator randomly selects couples of candidates that will be merged into one new candidate. Each candidate will be used for this crossover, resulting in 23 new candidates, and the process is performed twice so that the size of the total population after crossover is exactly 4 times that of the original (46 $\rightarrow$ 184). Note that candidates matched for merging are not necessarily disjoint. Since an age can not be present in an age group twice, the size of the merged candidate is not always the sum of the sizes of its “parents”.

Together, the mentioned operators provide a wide range of possibilities for improvement, providing fast growth of successful age groups through crossover, slight increases in size for good solutions and slight mutations within a fixed group for solutions that approach perfect.

Fitness Function

The selection method chosen to judge the fitness of all individuals in the population is two-fold. First there is a hard demand, stating that all age groups must show a significant difference between the control and offender group. Second, the larger the groups are, the more accurately, and according to Occam's Razor, the most plausibly they describe the online predator.

This selection is realized by assigning a bonus $\mathcal{B} = 10$ to an age group if its significance property has reached a certain threshold. This way, candidates that satisfy the significance demand are almost always ranked fitter than groups that do not.

Significant Difference

A significance test is a formal procedure for comparing observed data with a hypothesis, whose truth is to be assessed. The results of such a test are expressed in terms of a probability that measures how well the data and the hypothesis agree. Usually, the difference between two populations on some variable is calculated by assuming such a difference does not exist, called the *null-hypothesis* or $\mathcal{H}_0$, followed by a calculation that significantly falsifies that assumption, investigating the strength of the evidence against the hypothesis in terms of probability. The computed probability that denotes the chance that $\mathcal{H}_0$ is true is called the *p-value*. The smaller the p-value, the stronger the evidence against $\mathcal{H}_0$ is [32]. A number of different techniques exist that perform the calculation of this value, the *Student's t-test*, or "t-test" for short, being the most common. If the p-value is smaller than a certain significance level, denoted by α, which is usually set at 0.05, the groups are said to vary significantly on this variable.

There are a number of different t-tests of which the t-test for the comparison of two independent samples is applicable in our approach. In this test, the means ($\overline{x}_c$ and $\overline{x}_o$) and standard deviations (s_c and s_o) of the sample populations are tested against the respective theoretical means (μ_c and μ_o) and standard deviations (σ_c and σ_o) of their entire population, where c and o denote the control and offender group respectively. Through inference on $\mathcal{H}_0 : \mu_c = \mu_o$ a t-value is calculated as follows:

$$t = \frac{\overline{x}_c - \overline{x}_o}{\sqrt{\frac{s_c^2}{n_c} + \frac{s_o^2}{n_o}}},$$

where n_c and n_o are the respective sample sizes. This t-value effectively denotes the amount of theoretical standard deviations $\overline{x}_c$ lies from $\overline{x}_o$. This t has a *T-distribution*, a bell-shaped, 0-centered distribution that approaches the normal distribution when its "*degrees of freedom*" (*df*), approach infinity. This number is usually set to $n_c + n_o - 2$. The T-distribution lies just above the normal distribution in the outer corners, hence the more degrees of freedom, the lower the distribution will lie, the greater the chance that the p-value be lower than α, significantly rejecting $\mathcal{H}_0$. The calculation of the p-value is shown in Figure 10.8.

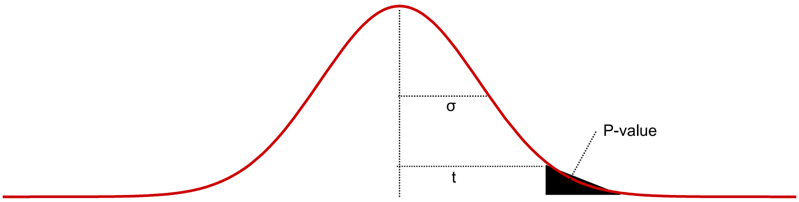


Figure 10.8: Calculating the P-value based on t

The p-value can be calculated by integrating on the t-distribution, with parameter df. The specification of this function falls outside the scope of this chapter but can be used as follows where $\mathcal{T}$ is the p-value reached through the t-test and $tdist$ is the integrated function for the t-distribution with df degrees of freedom:

$$\mathcal{T} = tdist(t, df)$$

A difficulty that arises with the usage of the t-test is that it assumes *normality*, meaning that the populations are normally distributed on the variable under investigation. This normality is said to hold in practice if the variable is distributed normally or if both sample population sizes are at least 30. Within our approach, the first demand is certainly not satisfied, while the second demand only holds for some of the larger age groups. Consequently, we have to resort to an alternative when either n_c or $n_o < 30$.

One of these good alternatives is the *Mann-Whitney* test (MW). This is a robust method that strives to compute a p-value for all populations, even those with a small size. Naturally, some accuracy is sacrificed reaching this objective. MW is based on quantitatively ranking every person in the sample based upon the variable under observation, followed by calculating the sum of ranks ($\mathcal{S}_x$) and average rank ($\overline{\mathcal{S}}_x = \frac{\mathcal{S}_x}{n_x}$) per group ($x = c$ or $x = o$). If $\mathcal{H}_0 : \mu_c = \mu_o$ holds, it is to be expected that $\overline{\mathcal{S}}_c \approx \overline{\mathcal{S}}_o$.

The test involves the calculation of a statistic, usually called $\mathcal{U}$, whose distribution under $\mathcal{H}_0$ can be approximated by the normal distribution. This is done by comparing the realized sum of ranks for one group ($\mathcal{R}_x$) with its maximal value:

$$\mathcal{U}_x = n_c n_o + \frac{n_x(n_x - 1)}{2} - \mathcal{R}_x,$$

where x is either c or o. Then

$$\mathcal{U} = \min(\mathcal{U}_c, \mathcal{U}_o)$$

Now z, the mount of standard deviations both group's mean differ, comparable to t from the t-test, can be computed by

$$z = \frac{(\mathcal{U} - \mu_U)}{\sigma_U}$$

μ and σ being the mean and standard deviations respectively, where, supposing $\mathcal{H}_0$ is true

$$\mu_U = \frac{n_c n_o}{2}$$

and

$$\sigma_U = \sqrt{\frac{n_c n_o (n_c + n_o + 1)}{12}}$$

The p-value can be calculated by integrating on the normal distribution. If $\mathcal{Z}$ is the p-value reached through the Mann-Whitney test and *ndist* is the integrated function for the normal distribution:

$$\mathcal{Z} = ndist(z)$$

Combining both methods in such a way that the strengths of both are used on applicability, statistical significance is computed as follows:

$$\mathcal{P} = \begin{cases} \mathcal{T} & \text{if } n_c \geq 30 \text{ and } n_o \geq 30 \\ \mathcal{Z} & \text{otherwise} \end{cases}$$

Combined with size

Now that the significance of the difference in amount of under-aged friends is computed, the group size variable can be added to calculate the fitness of an age group. A simple, yet effective, way to denote the size of a group is to count the number of single ages present (n_g). Measuring group size in individual ages also matches well with the computation of significance, which will be normalized between 0 and 10. Now, the fitness $\mathcal{F}$ of an age group is represented by:

$$\mathcal{S} = \begin{cases} \mathcal{B} + 10 \cdot \frac{\alpha - \mathcal{P}}{\alpha} & \text{if } \mathcal{P} \leq \alpha \\ 0 & \text{otherwise} \end{cases}$$

$$\mathcal{F} = n_g + \mathcal{S},$$

where $\mathcal{S}$ is the computed significance. This function yields a fitness between 2, being the minimum group size, and $\max(n_g) + 20$.

After applying the fitness function to all candidates, the population is ranked accordingly. Only the best quarter of the population is kept as the new generation, resulting in again 46 candidates.

The genetic algorithm presented above can end (successfully) in two ways. Either the average fitness has not increased for a set number of generations, most often signifying that no “child”-candidates are created that perform better than their “parents”, or a maximum number of generations is reached without entering such a state, after which the best candidates are selected anyway. This last number is usually reasonably large, ranging from 100,000 to 1,000,000 depending on the problem at hand.

10.4.2 Greedy Group Selection

After the GA finished, there are 46 groups remaining that have the potential of realizing the goal of discriminating between the two groups based upon age and percentage of under aged friends. We propose a greedy selection method that extracts the best candidates for this task from the final generation. There are six steps in this process

1. Sort candidates on fitness,
2. Start at the best candidate,
3. Scrap the candidate if it does not show a significant difference,
4. Select the candidate if it is disjoint to all previously selected age groups,
5. If not disjoint, cut the intersection and re-evaluate and resort all items ranked below including this item,
6. Go to next, lower ranked candidate and continue at step 3.

The first step is a logical step to take since significance is the goal to be reached, but after a large number of evolution cycles this step will be used very infrequently.

As is illustrated in Figure 10.9, an overlap can occur between two different age groups in the final selection. In this Figure, 5 individual age groups are shown, combined into two larger age groups by the GA, overlapping at the middle age. As step 5 dictates, the group with the lower fitness score surrenders the intersection. There are two reasons for this. First, as was demonstrated in Section 10.4 there can be a different boundary or threshold (Section 10.4.3) between control and offender group which prohibits a merger of the two if such a situation arises. If such a merge would have been possible, chances are that it would have happened during the GA phase of the approach. Second, the age group selected first is preserved, because for some reason it had a higher fitness than the second group. This can either be because it shows a more significant difference, is larger or both. In all these situations preservation of the stronger candidate is the preferred option.

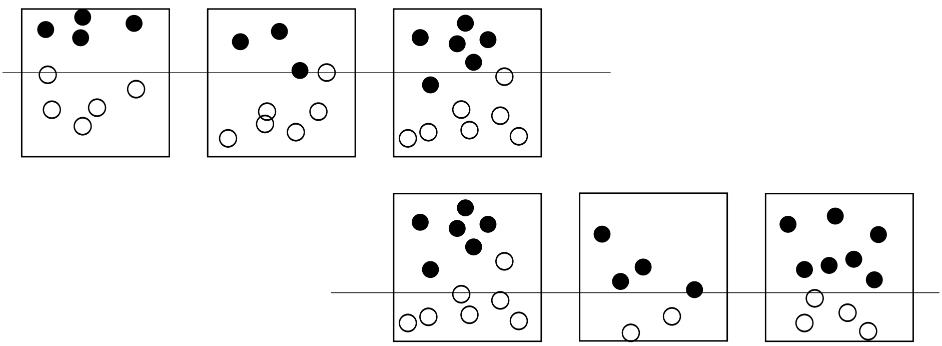


Figure 10.9: Two overlapping age groups, with different thresholds

Naturally, the size of the second group will decrease, reducing its fitness. One might also consider the chance that its significance could potentially be lowered, however, chances of this happening are extremely slim. Due to the fact that both groups were not merged in the GA phase, it is safe to assume that their boundaries differ. Hence, removal of the overlapping age will only set the threshold even further away from the first group, more closely approximating the boundary for the remaining ages and removing potentially overlapping percentages between the ages in the group. This will only lead to a higher significance level. Eventually, the fitness of the second individual does not necessarily drop greatly due to this modification.

10.4.3 Thresholding

Now that we have selected the discriminating age groups with the most descriptive value, they can be employed for classifying individuals in danger categories. Before this can be accomplished a threshold C_g must be designated for each age group g that denotes the decision line above which an individual is classified as potential offender and under which can be seen as safe. There are a number of different ways of constructing such thresholds, from which we choose the following four:

- Least false positives,
- Least false negatives,
- The average of the above.
- The weighted average of the above.

If the first option is applied, the method operates on the safe side, preferring to miss a predator, rather than falsely including a regular user. The second option can be used if one hopes to detect all predators, falsely classifying some regular users in the danger group. Taking the average of both could provide a more balanced view of reality, allowing for some false positives and negatives. The weighted average shifts the line more toward the least false negatives line if there were relatively many offenders in this age group and vice versa. The rationale behind this is that the more individuals were present in a sample the more representative that sample is for the entire population. It is calculated as follows:

$$\text{weightedaverage} = \frac{\frac{n_o}{n_c} b_{\text{lfn}} + \frac{n_c}{n_o} b_{\text{lfp}}}{2},$$

where b_{lfn} and b_{lfp} are the thresholds calculated by option 2 and 1 respectively.

It can also be the case that both groups are separated strictly, which occurs quite often if our hypothesis is correct. If this holds for a certain age(group), the method of setting the threshold is less relevant, but an average of the first two options is still preferred for clarity reasons.

Also, there are two options of setting these thresholds, by group or by every age from the groups separately. The first has the advantage that a clear image is communicated about the nature of the predator, while the second is more accurate, reducing the need for approximating over a larger number of samples.

It might be interesting to investigate how the area between b_{lfn} and b_{lfp}, that we will call the *ribbon*, is populated. Calculating the division of offenders and control, and the total amount of items in the ribbon can indicate how many overlap there is between the two samples. The ribbon $\mathcal{R}$ is calculated by:

$$\mathcal{R} = b_{\mathrm{lfn}} - b_{\mathrm{lfp}}$$

If there is any overlap between the sample populations the ribbon size will be negative, since the “no false positive”-line will lie under some of the samples in the control group. Therefore, a positive size of the ribbon is desirable, its size depicting the confidence of the existing significance.

The thresholds chosen within this phase can potentially be used to interpolate (for example through linear regression), to set a threshold for all ages, even though they were not shown to have a significant difference. This could both validate the exclusion of these ages, if a low accuracy is reached during testing and provide a function that can be used in future applications reducing storage needs of the outcome of this research.

10.5 Experimental Results

Our approach was tested on the national database of the Child Pornography Team of the Dutch National Police (KLPD) and a randomly drawn sample population of Hyves-users. We used all standard settings for our variables: $\mathcal{B} = 10$, $\alpha = 0.05$, $df = n_c + n_o - 2$ and set the maximum number of evolution cycles to 100,000.

Figure 10.10 displays the age groups that were discovered with the proposed approach. They are drawn at the weighted average threshold over entire age groups. In total, 4 different age groups were discovered, of which 3 were consecutive and 1 consisted of two separate groups. Together they contain 60.4% of all ages. Naturally, this graph has a similar form to the graph in Figure 10.4.

During the ten-fold cross-validation we computed the accuracy of our approach with all the different thresholding settings specified above. It was calculated by dividing the number of errors made in classification (both false positive and false negatives) by the number of correct classifications and subtracting that from 100%. Table 10.1 shows the results for the thresholding settings, where *LFP* and *LFN* denote *Least False Positives* and *Least False Negatives*, respectively.

Table 10.1: Accuracy results for all types of group thresholding

	LFP	*LFN*	*Average*	*W. average*
Accuracy	78%	81%	89%	92%

As discussed in Section 10.4.3, classification thresholds can also be set for individual ages instead of entire discriminating age groups. This could potentially lead to a higher accuracy. It also opens the door to interpolation of the results through linear regression. Figure 10.11 shows the results from this endeavor.

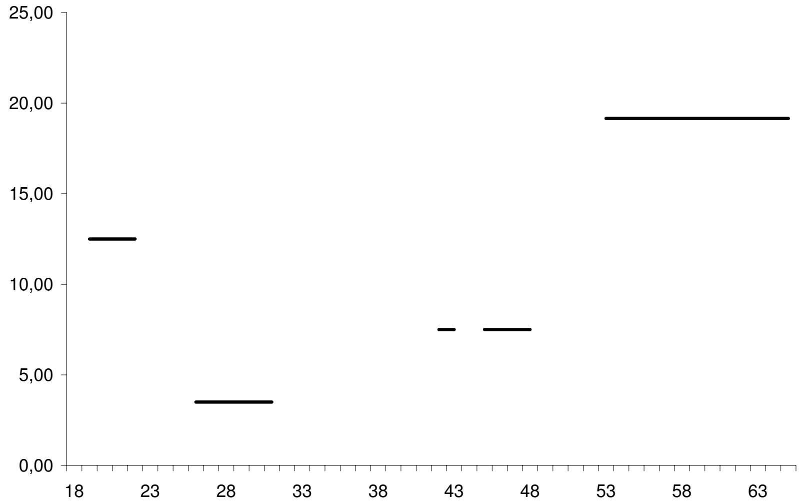


Figure 10.10: Discriminating age groups drawn at threshold height

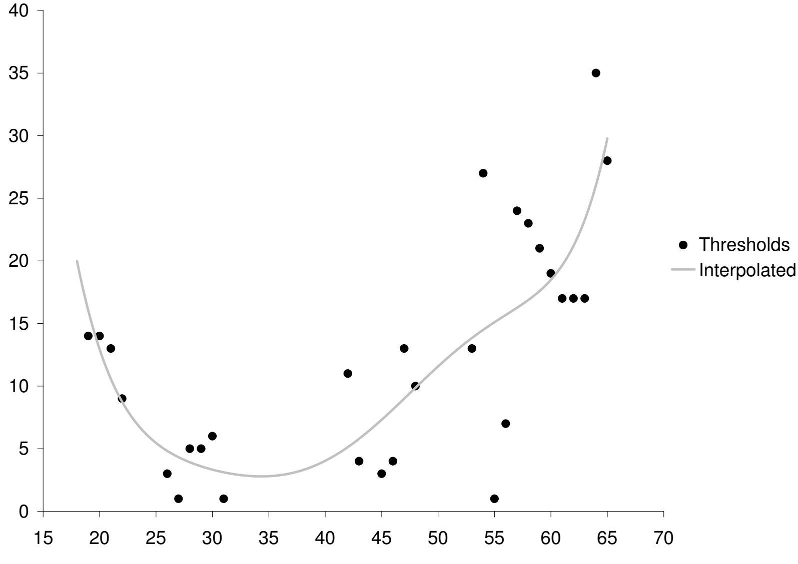


Figure 10.11: Individual thresholds supporting an interpolation line

It is clear from the graph that there is still a large variance in thresholds between the different "members" of an age group. This leads to the assumption that even better accuracy can be reached if the individual members are treated separately. If thresholding is done on an individual basis the results of an accuracy test differ quite a Table 10.1, as is obvious from Table 10.2. Also, now that the thresholds are points in the graph, an interpolation could be performed that sets thresholds based upon a polynomial, which yields results that are also presented in the table.

Table 10.2: Accuracy results for all types of individual thresholding

Discriminating groups						Other
	LFP	*LFN*	*Avg.*	*Wei.*	*Int.*	*Int.*
Acc.	83%	84%	95%	98%	82%	52%

The small size of the ribbon can be clearly observed in both the group thresholding and the individual thresholding case. It is 0.28 and 0.39 percentage points on average respectively. The items in the ribbon are 69% individuals from the offender sample.

10.6 Conclusion and Future Directions

The test shows a surprisingly large percentage of ages that actually support a significant difference between the offender and control samples. For these groups the best way to set a classification threshold appears to be based upon individual ages within this group. A weighted average of the "least false positives" and "least false negatives" extremes yields an accuracy of 98%, which suggests the method is a reliable tool to predict if a certain profile holder on a Social Networking Site is an online predator.

Unfortunately, at the same time, it appears to be impossible to draw any conclusion about people falling out of the discriminating age groups, which is only 40% of the population. An accuracy of 52% is too low to classify anything with any certainty. It is unfortunate that the preponderance interval of predators (30–42, Section 10.2), falls in this undecidable interval for 70%. However, the fact that no classification can be done after interpolation suggests these ages were left out by our method correctly, having no discriminative features. Interpolation is also to be avoided because it aversely affects the accuracy within the discriminating age groups.

The fact that the ribbon is very small suggests that the significance of the difference reached within the age groups barely holds. On a positive side, the number of offenders in the ribbon is substantially larger than the number of individuals from the control group. This indicates that our method classifies more false negatives than false positives, which is a good thing in most judiciary systems, supporting innocence until proven otherwise.

Now that these discriminating age groups have been established, they can be efficiently used within monitoring or early warning systems. Only 23 combinations of age and percentage need to be stored in order to do the classification.

Future research should mainly focus on two things: first, the outcome of our approach should be tested on other SNS, which can be easily done depending on the availability of police data, and second, the approach could be tested for other variables than age and percentage of under aged friends. The latter can be used to validate the outcome of our approach and could potentially shed more light on the age groups that were undecidable in our tests.

A last difficulty we ran into is the fact that not all queries can be executed via the openly accessible Hyves website. This means that for a detailed search query access to the full Hyves database is needed which is not possible for the ordinary Hyves user. After consultation, Hyves decided not to approve usage of its database directly, other than by official police demand, which we did not have. This severely limited a final test, that could have shown numbers of detected potential predators that were not on our shortlist or revealed profiles that we encountered in the construction of that list that were rejected because of doubts. If privacy legislation allows it (see Appendix A), future work could aim at acquiring this permission or extracting this information from the Hyves site by using web spiders in order to perform a final test that can provide a complete test case and an actual risk analysis on the presence of predators on SNS.

As in some of the previous chapters, there are some privacy issues and judicial constraints to the applicability of the methods discussed in this chapter. They matters are discussed in detail in Appendix A.

Appendix A

Statistical Significance and Privacy Issues

An intrinsic issue with the approaches described in this thesis is that there is a significant privacy and reliability issue, especially when they are applied on data as sensitive as that collected in the process of law enforcement. Applicability in daily police practice depends on the statistical insightfulness and relevance, current law and privacy regulation. In this chapter we describe some of the features of our approaches in this context and elaborate on their value in the application domain of law enforcement.

A.1 Statistics

A first important notion in the discussion of statistical issues surrounding our data mining approaches is that most of our efforts involved an *unguided search*, meaning that there was no specific occurrence or relation that we directed our searches toward. An advantage to this way of examining raw data is that more and especially potentially unexpected results are retrieved more easily than in a directed, in-depth statistical analysis. An intrinsic drawback to these kind of approaches, however, is that its results are founded on some statistical occurrence with a certain reliability, but are not able to, nor strive to, provide an explanation concerning the existence of the discovered patterns and in particular not on any underlying causality.

Due to this nature of the methods that are the algorithmic core of our approach, our tools and the accompanying results are less suitable in the research area of the social sciences, where insights into the background, existence and causality in the emergence of certain patterns are the main objective of scientific efforts. However, algorithms like the ones described in this thesis have been applied within a large number of different application domains, mostly in the form of decision support systems. Although one of the most usual deployment of these techniques is in the retail industry, both for companies and their customers, they are applicable in almost any area where decisions are to be made and where large amounts of data are available to have computer guided statistical data assistance. As is shown in the Introduction, some of these applications are currently used in the area of law enforcement, even though the principles underlying the emerging patterns are not revealed by the tools themselves. In cases where such an explanation is needed, the data resulting from the unguided search should, and is usually examined by a domain expert.

A good example where computer guided statistical data analysis is successfully implemented in the domain of law enforcement is the detection of fraudulent monetary transactions. Unguided searches through large database of transactions have been used widely in the process of discovering money laundering activities or tax evasion. If a pattern emerges, it is examined by a domain expert and appropriate action is taken according to his or her findings.

Below we provide a chapter wise discussion of the statistical applicability of the respective tools, that are an important background when the systems are to be put into actual use.

In Chapter 2 we describe a tool that yields relations (patterns) between the co-occurrence of crimes and demographic information. In this case, the pattern frequency is denoted by the percentual co-occurrence of its member in the data set. The novelty in this chapter resides in the way that these patterns are retrieved rather than a difficult statistical construct. The emerging patterns, accompanied by their frequency, can therefore be taken "as they are", describing a purely factual statement of occurrence. The retrieval methods do not explain why these patterns emerge, but the patterns themselves can be used to shed light on other existing phenomena within crime. The statistical means to acquire the patterns are widely known and applied. The same situation is applicable in Chapter 6, where the main focus is on the fast construction of the frequent subcareers, but the quality of the results is measured by a simple count of occurrences.

The visualization methods discussed in Chapter 3 are a good example where immediate feedback is provided about the statistical error that is made during the image construction process. The user is constantly informed through color coding how reliable the constructed visualization is, by inspecting the global error. This method is also used as an integral part of the approach in Chapter 4, where the visualization is the main result of the tool, accompanied by a clustering underlying it.

The distance measures discussed in Chapter 5 are all widely applied and their respective reliabilities and likelihood ratios are well-known. They are already used in many areas where reliability and robustness is of the essence, e.g., medical sciences, forensic research, however, the visualization method is quite different from the one used in Chapter 4. Therefore, a graph displaying the global error is displayed within the tool to inform the user of the quality of the constructed visualization (and therefore its underlying clustering). This is also the case for the construction of the visualization in Chapter 9. In that chapter the calculation of the distances is based upon the regular normal distribution.

In Chapter 7 and Chapter 8 a way of predicting the continuation of criminal careers is discussed. Since the possible outcome of such tools is highly sensitive we calculate two reliabilities. The first reliability describes how many times the prediction yields an accurate response in all cases known up to the current moment, providing users with a "general" reliability of the method. The second calculation is used to provide a reliability to a single prediction currently under consideration. The error made in the reliability calculation is 0.1, which is seen as very reliable. The same accuracy is reached in Chapter 10 if the most strict rules are used for set-separation.

In general, our methods are all subject to errors, but this error is assessed and made available to the user in every case. Especially since our tools do not describe the underlying principles for the patterns they yield, we encourage every possible user to be familiar with the statistical material described in the respective chapters of this thesis and familiarize themselves with the figures describing the reliability of the outcome.

A.2 Privacy

Naturally, approaches such as in this thesis come with privacy concerns. Authorative works on the application of data mining in the area of law enforcement in the Netherlands (our application domain, both through residence and project cooperation) are [37, 36], that discuss most relevant issues in law.

Every time data is analyzed for the purpose of law enforcement, the personal privacy is violated in a formal sense. According to international and national law, this is only acceptable if sufficient judicial principles exist that are known and accessible by civilians. It is important that laws are exact, so that civilians know beforehand when and how law enforcement agencies are allowed to use their authority in this sense. From this point of view, there are two main principles:

- Data on unsuspected individuals should only be analyzed in situations that are highly exceptional.
- Data on individuals should only be analyzed for the purpose it was collected for.

In the Netherlands these matters are arranged in *WPolr*(Law Police Registers) as a special case of the *Wbp* (Law Protection Personal Data). Important aspects in the application of these laws are the distribution of data and the principle of necessity. According to the first, the only police registers that can be distributed and used for most police tasks are the general police registers. According to the latter, every gathering, storage and analysis of data should be necessary for the police task. Each police register should have attached rules about how these principles are met [37]. The database and its current usage [29], described in Appendix B, comply with these demands.

Most of our methods are in complete accordance with these laws since they are drawn from existing police registers, meant for statistical analysis of trends in crime and should and can only applied in the area of crime analysis.

However, the tools described in Chapter 7 and Chapter 8 deal with the translation of newly discovered patterns or descriptions to individual cases rather than a general quest for truth or the discovery of general knowledge. This approach is best compared to the process of discovering suspicious financial transactions like for example in [23]. These methods also enforce criteria from a knowledge discovery support system on individual financial transactions in order to find outliers that might indicate for example money laundry activities. Naturally, not all transactions that yield warnings are actually fraudulent, but as long as the expected chance a warning is actually correct is reasonably high, the usage of such an decision support system is warranted as long as each warning is reviewed by experts before action is undertaken. The same should apply to the usage of our approach. However, in contrast with, among others, the above mentioned financial monitoring systems, that monitor *all* transactions done by *everybody*, only the people who have been active criminals for more than three years are under surveillance and even then, they are only electronically processed every time they commit a new crime.

However, our approach and the search for fraudulent transactions differ in one thing. The necessity principle has already been established for the latter, while no such directive has been determined for our tools. It is questionable if the necessity for suspect monitoring on possible criminal careers currently exists, especially since it is unknown if crime can be decreased though usage of our tools. Therefore, our tools in this matter are probably currently not allowed in the area of law enforcement, nor are they expected to be in the near future.

In Chapter 10 we describe a tool that uses publicly available data, to determine if there are aspects to users of Social Networking Sites that make them fall into danger categories for online predatation. It may however serve its purpose as a monitoring tool for larger internet applications, identifying profiles that may pose a risk to children and could be reviewed by a detective, but the possibilities in this area heavily depend on judicial constraints within the country of origin or application.

In the Netherlands, such a usage of this tool probably violates the first principle of the WPolr; our tool gathers data on unsuspected individuals, without the existence of a highly exceptional situation. Even though the data is publicly available and the data is not necessarily stored in a (semi-)permanent police register, the data is processed and therefore the usage of the tool in such a way is probably currently not possible or warranted.

However, in contrast to the more active approaches described in that chapter, the method has also potential as a strategic tool, helping to chart the dangers of predators on Social Networking Sites that could influence future legislation or police priorities.

Within this thesis we described methods that show the potential for computer guided statistical analysis on police data. However, usage and applicability are regulated by law, daily police activities and social necessity. Therefore, potential users are advised to familiarize themselves with the law in these matters before proceeding with the research in this thesis. Since in a chain of command, the situation might exist that executive personnel is not familiar with legislation concerning their activities and management personnel is not aware of the exact nature of the tasks their subordinates perform, the tools discussed in this thesis should provide proper warning of the privacy sensitive nature to any potential user, that might not be familiar with legislative concerns on their usage.

During our research no "new" potential offenders were identified nor put in any list that is currently monitored by police officials. Also, all data used within the research was either publicly available on the internet or made available by the police in an anonymized version. Hence, we assume no privacy sensitive footprints of our research remain.

Appendix B

The HKS database

A number of chapters deal with methods that are either designed to work with or tested on a database of criminal activity, describing offenders and their respective crimes. Throughout our research, we used the National HKS database of the Dutch National Police as a representative example. In this appendix we describe the structure of this database and the terms of use we observed during the testing phases of our research.

B.1 Structure

The Dutch National Police (Korps Landelijke Politie Diensten, KLPD), through its National Detective Force Information Department (Dienst Nationale Recherche Informatie, DNRI), annually extracts information from digital narrative reports stored throughout the individual, regional, administrative departments, and compiles this data into a large and reasonably clean database that contains all suspects and their possible crimes from the last decade. Since it is drawn from data stored in the individual HKS (Recognition systems, HerKenningsdienstSystemen), it is referred to as the National HKS Database.

Through a join on the municipal administration number (Gemeentelijke Basis Administratie, GBA) of the suspect, demographic information from Statistics Netherlands (Centraal Bureau voor de Statistiek, CBS) was added to the National HKS, for example including the age a person first committed a crime, his/her nationality and (ethnic) descend, and the perpetrator's gender [7, 36, 29].

In the main tables of the National HKS there are approximately one million rows, representing suspects and approximately 50 columns representing offenses and demographic data. All crimes are stored in a separate table that contains more details on this event (i.e., crime date, type) and can be linked to the individuals in the larger table. Crimes of an individual are split up into eight different types, varying from traffic and financial infringements to violent and sex crimes. All these crimes have got an intrinsic seriousness attached to them.

The compilation of the database started in 1998 and therefore only contains suspects and their potential offenses that have been reported within of after this year. However, every individual that is reported in this period is stored in the database with all its “baggage”, meaning that all potential offenses for this individual are present in the National HKS, even if they were reported before 1998.

The National HKS is stored in two distinct forms: the unmodified database as compiled from the individual departments and CBS, and an anonymized version where all identifiable information on individuals is omitted and replaced by a unique “suspect-number”.

B.2 Ownership and Residence

The ownership of the National HKS is rather complex and subtle, for the most part because there are many original sources, e.g., all Dutch municipalities and all local Dutch police departments, but for all practical reasons administrative ownership lies with both CBS and KLPD, where the latter is responsible for and owner of the characteristic part, e.g., the data on crimes. The ownership of data is depicted in Figure B.1

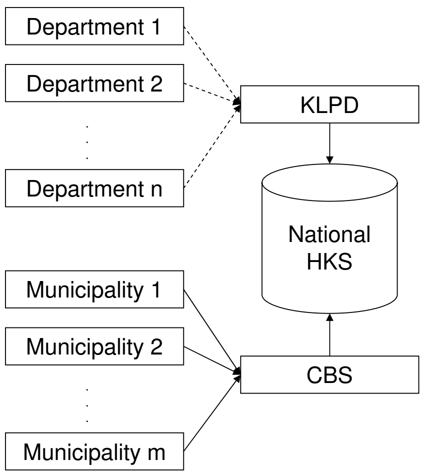


Figure B.1: Ownership of the National HKS

The actual situation, however, is somewhat more complex. Although original ownership on the narrative reports stored in the HKS lies with the individual police departments, they are not allowed to store data on suspects after a certain expiration date that is based upon closure of the case or some preset time frame set by the law (cf. Appendix A). The KLPD, however, is authorized to backup the data for a longer time if they restrict its usage to the general study of crime, its development in the country and offenders. By principle, the regional departments (or the KLPD) is not allowed to use any data on specific individuals that reside in the National HKS but is no longer available in any separate, regional HKS. Therefore, the ownership of the compiled data from regional departments can no longer be said to reside with these departments when the original data has been deleted from their systems, but is transferred to the KLPD, denoted by the dotted lines in Figure B.1.

There are two main locations where the National HKS is stored and accessible for usage: the headquarters of CBS and the Zoetermeer location of the KLPD, under auspices of its DNRI department. The copy at CBS is, as a principle, accessible by all who want to study it for scientific purposes, but only resides at their location in the anonymized version. The KLPD copy, on the other hand, is only accessible by KLPD personnel or by special request that is well motivated, especially since they manage both the original and anonymized version. The demographic data portion of the National HKS is available without any restriction in both locations.

B.3 Common Usage

There are two main objectives for the compilation of the national HKS: the possibility to answer general questions on the nature of or trends within crime to law enforcement officials and the compilation of the Nation-wide Crime Chart (Landelijke Criminaliteits-Kaart, LCK [29]). The first objective is met by the ability to contact the DNRI by phone.

The LCK is an annual report, fabricated by DRNI, that describes a wide variety of data and knowledge on and trends within crime and observes a large number of trends within the crime area, providing ample insights into the relations between crime, city-size, ethniticity, law enforcement policy and police deployment.

Although they are consistently referred to as suspects, all individuals are treated as offenders in these publications, as can be derived from nomenclatures like "recidivism", "criminal foreigners" and "criminal" profiles, and all narrative reports are considered to describe events that really happened. No checking is performed on the outcome of police investigations or court proceedings, nor are they considered relevant in the pursuit of the objective to provide insights into the existence and development of crime.

Apart from the two owning organizations, research on the National HKS is also done by the regional police departments, ministries, the Netherlands Court of Audit (Algemene Rekenkamer), municipalities and research organizations, mostly in the anonymized form.

B.4 Usage during Research

During the research underlying this thesis, we performed a number of tests on a collection of meta-data derived from the National HKS. For this purpose we obtained permission from the KLPD to use their copy of the anonymized version, provided the database remained at KLPD headquarters.

Before the data was analyzed by our tool it was transformed into numbered series of integers that represent criminal careers, but were untraceable to the original data. Two steps were taken within this encoding:

- **No numbering on suspects** Since our research does not deal with any individual suspects, nor do we need information uniquely identifying them, all unique numbering was removed.

- **Offset all time frames** Since we are not interested in data on actual time frames, relating certain events to certain real life periods, we offset all data on this to start at time frame 1, identifying the first time frame of activity that is increased with 1 for each next time frame.

An example of this can be viewed in Figure B.2, where the numbers represent specific crime types.

Year **Suspect**	**2002**	**2003**	**2004**	**2005**
23		15 15	38	15 38
25	15		29	

Transformed to

1:	1:	15
		15
	2:	38
	3:	15
		38
2:	1:	15
	3:	29

Figure B.2: Fictive example of the transformation of HKS data before analysis by our tools

As is clear from the figure, all relation between original suspects and numbered series are lost. On top of that, there is no longer a relation between actual years and the numbered time frames in the transformed list, especially since there is no relation between the first year of activity for suspect 1 and the first year of suspect 2 (both represent different years but are denoted by 1).

Just as in the official KLPD publications (like the LCK), we assumed all suspects were actual offenders and all reported events had actually happened. However, no conclusions were drawn about specific or retraceable individuals. Also, all data resulting from our research on the National HKS has been made available to the KLPD personnel attached to our project for further examination.

Bibliography for Part II

[1] H. Abdi. Least squares. In *Encyclopedia for Research Methods for the Social Sciences*, pages 792–795. Thousand Oaks (CA): Sage, 2003.

[2] H. Abdi. Signal detection theory. In *Encyclopedia of Measurement and Statistics*. Thousand Oaks (CA): Sage, 2007.

[3] American Psychological Association. *Merriam-Webster's Medical Dictionary*. Merriam-Webster, Incorporated, 2007.

[4] J. Anderson. *Learning and Memory*. Wiley and Sons, New York, 1995.

[5] R.H. Bartels, J.C. Beatty, and B.A. Barsky. *An Introduction to Splines for Use in Computer Graphics and Geometric Modelling*. Morgan Kaufmann, 1987.

[6] C.M. Bishop. *Pattern Recognition and Machine Learning*. Springer Science, 2006.

[7] M. Blom, J. Oudhof, R.V. Bijl, and B.F.M. Bakker. Verdacht van criminaliteit, allochtonen en autochtonen nader bekeken (English from p. 84). Technical Report 2005-2, Dutch, Scientific Research and Documentation Centre (WODC) and Statistics Netherlands (CBS), 2005.

[8] D.M. Boyd and N.B. Ellison. Social network sites: Definition, history and scholarship. *Journal of Computer-Mediated Communication*, 13(1), 2007.

[9] J. Broekens, T. Cocx, and W.A. Kosters. Object-centered interactive multi-dimensional scaling: Ask the expert. In *Proceedings of the Eighteenth Belgium-Netherlands Conference on Artificial Intelligence (BNAIC2006)*, pages 59–66, 2006.

[10] J.S. de Bruin, T.K. Cocx, W.A. Kosters, J.F.J. Laros, and J.N. Kok. Data mining approaches to criminal career analysis. In *Proceedings of the Sixth IEEE International Conference on Data Mining (ICDM 2006)*, pages 171–177. IEEE, 2006.

[11] J.S. de Bruin, T.K. Cocx, W.A. Kosters, J.F.J. Laros, and J.N. Kok. Onto clustering criminal careers. In *Proceedings of the ECML/PKDD 2006 Workshop on Practical Data Mining: Applications, Experiences and Challenges*, pages 92–95, 2006.

[12] M. Chau, J. Xu, and H. Chen. Extracting meaningful entities from police narrative reports. In *Proceedings of The National Conference on Digital Government Research*, pages 1–5, 2002.

[13] T.K. Cocx, W.A. Kosters, and J.F.J. Laros. Enhancing the automated analysis of criminal careers. In *SIAM Workshop on Link Analysis, Counterterrorism, and Security 2008 (LACTS2008)*. SIAM Press, 2008.

[14] T.K. Cocx, W.A. Kosters, and J.F.J. Laros. Temporal extrapolation within a static clustering. In *Foundations of Intelligent Systems, Proceedings of the Seventeenth International Symposium on Methodologies for Intelligent Systems (ISMIS 2008)*, volume 4994 of *LNAI*, pages 189–195. Springer, 2008.

[15] M.L. Davison. *Multidimensional Scaling*. John Wiley and Sons, New York, 1983.

[16] S.C. Dombrowski, K.L. Gischlar, and T. Durst. Safeguarding young people from cyber pornography and cyber sexual predation: A major dilemma of the internet. *Child Abuse Review*, 16(3):153–170, 2007.

[17] S.C. Dombrowski, J.W. LeMasney, C. E. Ahia, and S.A. Dickson. Protecting children from online sexual predators: Technological, psychoeducational, and legal considerations. *Professional Psychology: Research and Practice*, 35(1):65–73, 2004.

[18] J.S. Donath. Identity and deception in the virtual community. *Communities in Cyberspace*, 1998.

[19] A.E. Eiben and J.E. Smith. *Introduction to Evolutionary Computing*. Springer, 2003.

[20] M. Elliott, K. Browne, and J. Kilcoyne. Child abuse prevention: What offenders tell us. *Child Abuse and Neglect*, 19:579–594, 1995.

[21] FBI. The FBI strategic plan, 2004–2009, `http://www.fbi.gov/`.

[22] D. Finkelhor, K. Mitchell, and J. Wolak. Online victimization: A report on the nation's youth. Technical report, National Center for Missing and Exploited Children, 2000.

[23] H.G. Goldberg and R.W.H. Wong. Restructuring transactional data for link analysis in the FinCEN AI system. In *Papers from the AAAI Fall Symposium*, pages 38–46, 1998.

[24] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning; Data Mining, Inference, and Prediction*. Springer, 2001.

[25] W.A. Kosters and J.F.J. Laros. Visualization on a closed surface. In *Proceedings of the Nineteenth Belgium-Netherlands Conference on Artificial Intelligence (BNAIC2007)*, pages 189–195, 2007.

[26] W.A. Kosters, E. Marchiori, and A. Oerlemans. Mining clusters with association rules. In *Proceedings of the Third Symposium on Intelligent Data Analysis (IDA99)*, volume 1642 of *LNCS*, pages 39–50. Springer, 1999.

[27] W.A. Kosters and M.C. van Wezel. Competitive neural networks for customer choice models. In *E-Commerce and Intelligent Methods, Studies in Fuzziness and Soft Computing 105*, pages 41–60. Physica-Verlag, Springer, 2002.

[28] N. Kumar, J. de Beer, J. Vanthienen, and M.-F. Moens. Evaluation of intelligent exploitation tools for non-structured police information. In *Proceedings of the ICAIL 2005 Workshop on Data Mining, Information Extraction and Evidentiary Reasoning for Law Enforcement and Counter-terrorism*, 2005.

[29] J. Lammers, W. van Tilburg, L. Prins, H. de Miranda, and K. Lakhi. National criminal chart 2004 (Landelijke criminaliteitskaart 2004). Technical Report 27/2005, Dutch, Dienst Nationale Recherche Informatie (DNRI), 2005.

[30] L.A. Malesky. Predatory online behavior: Modus operandi of convicted sex offenders in identifying potential victims and contacting minors over the internet. *Journal of Child Sexual Abuse*, 16(2):23–32, 2007.

[31] K.J. Mitchell, J. Wolak, and D. Finkelhor. Police posing as juveniles online to catch sex offenders: Is it working? *Sexual Abuse: A Journal of Research and Treatment*, 17(3):241–267, 2005.

[32] D.S. Moore and G.P. McCabe. *Introduction to the Practice of Statistics*. W.H. Freeman and Co., 4th edition, 2003.

[33] T. O'Reilly. What is Web 2.0 — Design patterns and business models for the next generation of software. Technical report, O'Reilly, 2005.

[34] K. Pearson. On lines and planes of closest fit to systems of points in space. *Phil. Mag.*, 2(6):559–572, 1901.

[35] C. Runge. Über Empirische Funktionen und die Interpolation zwischen äquidistanten Ordinaten. *Zeitschrift für Mathematik und Physik*, 46:224–243, 1901.

[36] B. Schermer. *Software Agents, Surveillance, and the Right to Privacy: A Legislative Framework for Agent-enabled Surveillance*. PhD thesis, Leiden University, 2007.

[37] R. Sietsma. *Gegevensverwerking in het kader van de opsporing; toepassing van datamining ten behoeve van de opsporingstaak: Afweging tussen het opsporingsbelang en het recht op privacy*. PhD thesis, Dutch, Leiden University, 2007.

[38] M. Sullivan. *Safety Monitor, How to Protect your Kids Online*. Bonus Books, 2002.

[39] R. Sun and C.L. Giles. Sequence learning: From recognition and prediction to sequential decision making. *IEEE Intelligent Systems*, 16:67–70, 2001.

[40] R. Sun, E. Merrill, and T. Peterson. From implicit skills to explicit knowledge: A bottom-up model of skill learning. *Cognitive Science*, 25(2):203–244, 2001.

[41] SPSS LexiQuest website. `http://www.spss.com/spssbi/lexiquest/`, accessed 01-2006.

Acknowledgments

Although the creation of a PhD thesis is mostly an individual effort, there are some people that helped me out during my research or to whom I owe some gratitude for other reasons.

First of all I would like to thank the following people for making important contributions to the content of my peer reviewed papers, as co-authors or in another form, or this thesis: Jeroen Laros, Robert Brijder, Joost Broekens, Eric Kuijl and Kees Vuik.

I would like to thank NWO for giving me the monetary opportunity to work on this thesis and the people at the KLPD for their invaluable support to this project: Ton Holtslag, Henry Willering, Jochen van der Wal, Wil van Tilburg and Leen Prins.

During the work on this thesis I received mental support from a large number of people, most prominently the VTG group, HKA, the coffee break group and of course my parents Ben and Vera Cocx and my dear girlfriend Daphne Swiebel: thank you for your never ending interest in my work and my results, and your support when things were difficult.

If you do not find yourself on the list but have followed me through the years, know that your interest has not been in vain and has helped me in becoming the author of this work and more.

Nederlandse Samenvatting

Op de achtergrond van de data-explosie van de late jaren 1990 is er een onderzoeksgebied geëvolueerd uit de statistiek en informatica. Het hoofddoel van deze vorm van computergestuurde data analyse, die bekend staat als *data mining* (het graven in gegevens) of *Knowledge Discovery in Databases* (KDD) ("kennis ontdekking" in databases), is om kennis te extraheren uit een, vaak grote, collectie van "ruwe" data, waarbij elementen uit de statistiek, database technologie, kunstmatige intelligentie, visualisatie en uit het machine-leren worden gecombineerd. Een belangrijk aspect van het vakgebied is het omgaan met gegevens die niet specifiek ontworpen werden om er computergestuurde analyses op uit te voeren. Bij het bedrijven van data mining draait het meestal om het vinden van onverwachte patronen en andere verrassende resultaten waar niet direct of concreet op gezocht werd.

Het toenemen van de mogelijkheden in de informatietechnologie van het laatste decennium heeft geleid tot een grote toename van de hoeveelheid opgeslagen gegevens, zowel als een zij-product van bedrijfs- en overheidsadministratie, als resultaat van wetenschappelijke analyses. Hoewel de meeste van deze gegevens een inherent nut met zich meebrengen, zoals klant management, het archiveren van belastingteruggaves of het uitvoeren van DNA analyses, heeft data mining software als doel om kennis te aggregeren uit deze gegevens, door het automatisch opsporen van (onderliggende) patronen, gedragsklassen of communicatienetwerken. Respectievelijk kan zo waardevolle kennis worden ingewonnen over klant gedrag, belastingontduiking of over stukjes DNA die garant staan voor, biologische afwijkingen. Meestal kunnen de algoritmes, opgesteld voor dit soort taken betrekkelijk eenvoudig overgebracht worden naar andere expertise-domeinen.

Een van deze potentiële domeinen is dat van de wetshandhaving. Als een onderdeel van nationale of regionale overheden dat gebonden is aan strenge regelgeving, is de overgang van een papieren administratie naar een digitale informatie infrastructuur langzaam gegaan, maar de laatste jaren, zeker na de aanslagen van 11 september 2001, hebben politieorganisaties meer geïnvesteerd in specifieke, bruikbare en uniforme informatiesystemen. Zoals in alle andere gebieden, heeft dit proces geleid tot zeer veel gegevens, die waarschijnlijk zeer bruikbaar zijn voor data mining doeleinden. Dit proefschrift beschrijft een aantal projecten in deze richting en rapporteert de resultaten die bereikt werden door toepassing van de ontwikkelde algoritmes op daadwerkelijke politie gegevens. Hiermee wordt een basis gelegd voor een toekomst waarbij speciaal ontwikkelde algoritmische assistentie een waardevolle rol kan spelen bij het bestrijden van de misdaad.

Het gerapporteerde onderzoek in dit proefschrift is onderverdeeld in twee delen, die direct gerelateerd zijn aan twee vormen van de operationele wetshandhaving zoals die in Nederland bestaan: de strategische en de tactische wetshandhaving. Bij het eerste moet gedacht worden aan het uitzetten van (korps-)beleid met betrekking tot locaties, personeelsbezetting, et cetera, op basis van bestaande gegevens over criminaliteit. De tactisch geörienteerde activiteiten zijn vaak direct gerelateerd aan opsporing en zaak-analyses.

In Hoofdstuk 2, als eerste hoofdstuk van het strategisch deel, wordt een eerste analyse uitgevoerd van een grote database van strafbladen (zie Appendix B), waarbij gepoogd wordt verschillende misdaden aan elkaar of aan demografische gegevens te relateren, op basis van het feit dat ze vaak samen "voorkomen" in een strafblad. Om dit te bereiken wordt het bekende algoritme A-priori aangepast om in dit specifieke bestand te zoeken naar dergelijke verbanden, terwijl het rekening houdt met een variëteit aan problemen die inherent zijn aan criminaliteitsgegevens. Dit bestand, dat in geanonimiseerde versie beschikbaar werd gesteld, is eveneens gebruikt in andere analyses.

Omdat dit bestand een behoorlijke omvang "ruwe" data heeft, zijn standaard methoden om deze data te visualiseren vaak niet erg geschikt. In Hoofdstuk 3 wordt daarom een methode beschreven die het weergeven van verbanden tussen criminelen uit dit bestand optimaliseert door de domein kennis van de analist te betreken bij het "clusteren". Doordat deze expert direct de fysieke weergave van de gegevens kan manipuleren kan met een relatief lage "berekeningscomplexiteit" een zeer hoge kwaliteit visualisatie behaald worden.

Een belangrijk concept dat geëvalueerd kan worden met behulp van het strafbladenbestand is dat van de *criminele carrière*, dat beschouwd kan worden als een temporeel geordende serie misdaden die een individu tijdens zijn leven begaan heeft. Een ad-hoc methode wordt gesuggereerd in Hoofdstuk 4, waarbij vier belangrijke factoren van een carrière gebruikt worden om afstanden tussen verschillende carrières te berekenen. Deze afstanden kunnen vervolgens gevisualiseerd worden in een twee-dimensionale clustering. Hoofdstuk 5 stelt een aantal verbeteringen op deze methode voor die allemaal al functioneel zijn gebleken op een ander gebied. Eveneens worden daar methoden beschreven om uit deze gegevens nieuwe carrières te kunnen voorspellen, waarop verder ingegaan wordt in Deel II.

Nu een systeem ontworpen is voor het clusteren en classificeren van criminele carrìeres, is het mogelijk te zoeken naar vaak voorkomende subcarrières. Een nog belangrijkere onderneming is het zoeken naar subcarrières die vaak voorkomen in een specifieke klasse maar juist niet in alle anderen, zodat zij de rol op zich kunnen nemen van defini erende subcarrière, die gebruikt kan worden om bepaalde klasses te identificeren. Deze mogelijkheden worden behandeld in Hoofdstuk 6, waar een bestaande methode voor winkelmandjes-analyse wordt aangepast om tegemoet te komen aan de specifieke eisen voor de zoektocht naar subcarrières.

In het tweede deel wordt één van deze mogelijkheden beschreven in Hoofdstuk 7, via een methode om de kracht van een visualisatie aan te wenden om via simpele mathematische berekeningen een betrouwbare voorspelling te doen. Deze methode wordt uitgewerkt en speciaal toegespitst op criminaliteitsgegevens in Hoofdstuk 8, waar de verschillende variabelen van deze methodiek worden getoetst op daadwerkelijke data.

Via deze methode kunnen criminele carrières onder bepaalde omstandigheden met grote nauwkeurigheid voorspeld worden.

In Hoofdstuk 9 staat een onderzoek beschreven dat zich buigt over de vraag in hoeverre bestanden van in beslag genomen computers een indicatie kunnen zijn voor welke misdrijflocaties gerelateerd zijn aan dezelfde criminele organisaties. Voor dit doel werd een speciale afstandsmaat gedefinieerd die bepaald hoe groot de kans is dat twee computers bij dezelfde organisatie behoren. Hierbij werd gebruik gemaakt van tekstherkenningssoftware die tekst extraheerde van computers uit drugslaboratoria.

In Hoofdstuk 10 staat beschreven hoe *online predators*, "kinderlokkers op internet", automatisch herkend zouden kunnen worden op sociale netwerkomgevingen, zoals het Nederlandse Hyves. Er wordt beschreven hoe een genetisch algoritme automatisch groepen selecteert waartussen een significant verschil op te merken is tussen deze predators en andere gebruikers in het aantal minderjarige vrienden op hun profiel. Het blijkt dat deze variabele in sommige gevallen een zeer sterke indicator kan zijn voor risico classificatie van bepaalde gebruikersgroepen.

Dit proefschrift eindigt met Appendix A waarin een aantal overwegingen worden gegeven met betrekking tot statistiek, recht en privacy die een cruciale rol spelen voor iedereen die (een deel van) ons werk gebruikt of van plan is te gebruiken in de dagelijkse omgeving van het politiewerk. Het bespreekt de toepasbaarheid, statistische relevantie en inzichtelijkheid van en wat voorbehoudens met betrekking tot onze methodieken in het algemeen en voor politiegebruik in het bijzonder, waarbij vooral gefocust wordt op de gevoeligere toepassingen, besproken in Deel II. Om er zeker van te zijn dat onze methodes op de correcte manier bekeken en onze tools op een nauwkeurige en gepaste wijze gebruikt worden, is een behandeling van hun mogelijkheden en beperkingen voor maatschappelijk gebruik zowel belangrijk als vanzelfsprekend.

Curriculum Vitae

Tim Cocx is geboren in Amstelveen, Noord-Holland, op 18 december 1979. Van 1992 tot 1998 doorliep hij het Gymnasium op het Sint Nicolaas Lyceum te Amsterdam. Van 1998 tot 2004 studeerde hij Informatica aan de Universiteit Leiden, waar hij de laatste jaren ook betrokken was bij verschillende vakken als practicum- en werkgroepassistent. Vanaf 2005 was hij verbonden aan het Leiden Institute of Advanced Computer Science als promovendus, waar hij zijn promotieonderzoek uitvoerde. Eveneens gaf hij in die jaren verschillende vakken in de Masteropleiding Mediatechnologie.

Publication List

Most chapters in this thesis are based peer reviewed publications. Chapter 2 is based upon work published in [4]. In Chapter 3 we elaborate on research first published as [1]. Chapter 4 is a modification of [2] and was also published in extended abstract form [3]. The research described in Chapter 5 was first made public in [7]. Chapter 7, published as [8, 9], is a prelude to Chapter 8 that is based upon [6]. The author's first published paper [5] appears in this thesis as Chapter 9.

Chapter 10 was accepted for presentation at the AusDM conference, Australia, in a previous form and to appear in *Conferences in Research and Practice in Information Technology*, but was withdrawn due to organizational issues. Chapter 6 was written specifically for this thesis.

[1] J. Broekens, T. Cocx, and W.A. Kosters. Object-centered interactive multi-dimensional scaling: Ask the expert. In *Proceedings of the Eighteenth Belgium-Netherlands Conference on Artificial Intelligence (BNAIC2006)*, pages 59–66, 2006.

[2] J.S. de Bruin, T.K. Cocx, W.A. Kosters, J.F.J. Laros, and J.N. Kok. Data mining approaches to criminal career analysis. In *Proceedings of the Sixth IEEE International Conference on Data Mining (ICDM 2006)*, pages 171–177. IEEE, 2006.

[3] J.S. de Bruin, T.K. Cocx, W.A. Kosters, J.F.J. Laros, and J.N. Kok. Onto clustering criminal careers. In *Proceedings of the ECML/PKDD 2006 Workshop on Practical Data Mining: Applications, Experiences and Challenges*, pages 92–95, 2006.

[4] T. K. Cocx and W.A. Kosters. Adapting and visualizing association rule mining systems for law enforcement purposes. In *Proceedings of the Nineteenth Belgium-Netherlands Conference on Artificial Intelligence (BNAIC2007)*, pages 88–95, 2007.

[5] T.K. Cocx and W.A. Kosters. A distance measure for determining similarity between criminal investigations. In *Advances in Data Mining, Proceedings of the Industrial Conference on Data Mining 2006 (ICDM2006)*, volume 4065 of *LNAI*, pages 511–525. Springer, 2006.

[6] T.K. Cocx, W.A. Kosters, and J.F.J. Laros. An early warning system for the prediction of criminal careers. In *MICAI 2008: Advances in Artificial Intelligence*, volume 5317 of *LNAI*, pages 77–89. Springer, 2008.

[7] T.K. Cocx, W.A. Kosters, and J.F.J. Laros. Enhancing the automated analysis of criminal careers. In *SIAM Workshop on Link Analysis, Counterterrorism, and Security 2008 (LACTS2008)*. SIAM Press, 2008.

[8] T.K. Cocx, W.A. Kosters, and J.F.J. Laros. Temporal extrapolation within a static clustering. In *Foundations of Intelligent Systems, Proceedings of the Seventeenth International Symposium on Methodologies for Intelligent Systems (ISMIS 2008)*, volume 4994 of *LNAI*, pages 189–195. Springer, 2008.

[9] T.K. Cocx, W.A. Kosters, and J.F.J. Laros. Temporal extrapolation within a static clustering, extended abstract. In *Proceedings of the Eighteenth Belgium-Netherlands Conference on Artificial Intelligence (BNAIC2008)*, pages 295–296. BNAIC, 2008.

Titles in the IPA Dissertation Series since 2005

E. Ábrahám. *An Assertional Proof System for Multithreaded Java -Theory and Tool Support-* . Faculty of Mathematics and Natural Sciences, UL. 2005-01

R. Ruimerman. *Modeling and Remodeling in Bone Tissue*. Faculty of Biomedical Engineering, TU/e. 2005-02

C.N. Chong. *Experiments in Rights Control - Expression and Enforcement*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2005-03

H. Gao. *Design and Verification of Lock-free Parallel Algorithms*. Faculty of Mathematics and Computing Sciences, RUG. 2005-04

H.M.A. van Beek. *Specification and Analysis of Internet Applications*. Faculty of Mathematics and Computer Science, TU/e. 2005-05

M.T. Ionita. *Scenario-Based System Architecting - A Systematic Approach to Developing Future-Proof System Architectures*. Faculty of Mathematics and Computing Sciences, TU/e. 2005-06

G. Lenzini. *Integration of Analysis Techniques in Security and Fault-Tolerance*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2005-07

I. Kurtev. *Adaptability of Model Transformations*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2005-08

T. Wolle. *Computational Aspects of Treewidth - Lower Bounds and Network Reliability*. Faculty of Science, UU. 2005-09

O. Tveretina. *Decision Procedures for Equality Logic with Uninterpreted Functions*. Faculty of Mathematics and Computer Science, TU/e. 2005-10

A.M.L. Liekens. *Evolution of Finite Populations in Dynamic Environments*. Faculty of Biomedical Engineering, TU/e. 2005-11

J. Eggermont. *Data Mining using Genetic Programming: Classification and Symbolic Regression*. Faculty of Mathematics and Natural Sciences, UL. 2005-12

B.J. Heeren. *Top Quality Type Error Messages*. Faculty of Science, UU. 2005-13

G.F. Frehse. *Compositional Verification of Hybrid Systems using Simulation Relations*. Faculty of Science, Mathematics and Computer Science, RU. 2005-14

M.R. Mousavi. *Structuring Structural Operational Semantics*. Faculty of Mathematics and Computer Science, TU/e. 2005-15

A. Sokolova. *Coalgebraic Analysis of Probabilistic Systems*. Faculty of Mathematics and Computer Science, TU/e. 2005-16

T. Gelsema. *Effective Models for the Structure of pi-Calculus Processes with Replication*. Faculty of Mathematics and Natural Sciences, UL. 2005-17

P. Zoeteweij. *Composing Constraint Solvers*. Faculty of Natural Sciences, Mathematics, and Computer Science, UvA. 2005-18

J.J. Vinju. *Analysis and Transformation of Source Code by Parsing and Rewriting*. Faculty of Natural Sciences, Mathematics, and Computer Science, UvA. 2005-19

M.Valero Espada. *Modal Abstraction and Replication of Processes with Data*. Faculty of Sciences, Division of Mathematics and Computer Science, VUA. 2005-20

A. Dijkstra. *Stepping through Haskell*. Faculty of Science, UU. 2005-21

Y.W. Law. *Key management and link-layer security of wireless sensor networks: energy-efficient attack and defense*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2005-22

E. Dolstra. *The Purely Functional Software Deployment Model*. Faculty of Science, UU. 2006-01

R.J. Corin. *Analysis Models for Security Protocols*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2006-02

P.R.A. Verbaan. *The Computational Complexity of Evolving Systems*. Faculty of Science, UU. 2006-03

K.L. Man and R.R.H. Schiffelers. *Formal Specification and Analysis of Hybrid Systems*. Faculty of Mathematics and Computer Science and Faculty of Mechanical Engineering, TU/e. 2006-04

M. Kyas. *Verifying OCL Specifications of UML Models: Tool Support and Compositionality*. Faculty of Mathematics and Natural Sciences, UL. 2006-05

M. Hendriks. *Model Checking Timed Automata - Techniques and Applications*. Faculty of Science, Mathematics and Computer Science, RU. 2006-06

J. Ketema. *Böhm-Like Trees for Rewriting*. Faculty of Sciences, VUA. 2006-07

C.-B. Breunesse. *On JML: topics in tool-assisted verification of JML programs*. Faculty of Science, Mathematics and Computer Science, RU. 2006-08

B. Markvoort. *Towards Hybrid Molecular Simulations*. Faculty of Biomedical Engineering, TU/e. 2006-09

S.G.R. Nijssen. *Mining Structured Data*. Faculty of Mathematics and Natural Sciences, UL. 2006-10

G. Russello. *Separation and Adaptation of Concerns in a Shared Data Space*. Faculty of Mathematics and Computer Science, TU/e. 2006-11

L. Cheung. *Reconciling Nondeterministic and Probabilistic Choices*. Faculty of Science, Mathematics and Computer Science, RU. 2006-12

B. Badban. *Verification techniques for Extensions of Equality Logic*. Faculty of Sciences, Division of Mathematics and Computer Science, VUA. 2006-13

A.J. Mooij. *Constructive formal methods and protocol standardization*. Faculty of Mathematics and Computer Science, TU/e. 2006-14

T. Krilavicius. *Hybrid Techniques for Hybrid Systems*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2006-15

M.E. Warnier. *Language Based Security for Java and JML*. Faculty of Science, Mathematics and Computer Science, RU. 2006-16

V. Sundramoorthy. *At Home In Service Discovery*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2006-17

B. Gebremichael. *Expressivity of Timed Automata Models*. Faculty of Science, Mathematics and Computer Science, RU. 2006-18

L.C.M. van Gool. *Formalising Interface Specifications*. Faculty of Mathematics and Computer Science, TU/e. 2006-19

C.J.F. Cremers. *Scyther - Semantics and Verification of Security Protocols*. Faculty of Mathematics and Computer Science, TU/e. 2006-20

J.V. Guillen Scholten. *Mobile Channels for Exogenous Coordination of Distributed Systems: Semantics, Implementation and Composition*. Faculty of Mathematics and Natural Sciences, UL. 2006-21

H.A. de Jong. *Flexible Heterogeneous Software Systems*. Faculty of Natural Sciences, Mathematics, and Computer Science, UvA. 2007-01

N.K. Kavaldjiev. *A run-time reconfigurable Network-on-Chip for streaming DSP applications*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2007-02

M. van Veelen. *Considerations on Modeling for Early Detection of Abnormalities in Locally Autonomous Distributed Systems*. Faculty of Mathematics and Computing Sciences, RUG. 2007-03

T.D. Vu. *Semantics and Applications of Process and Program Algebra*. Faculty of Natural Sciences, Mathematics, and Computer Science, UvA. 2007-04

L. Brandán Briones. *Theories for Model-based Testing: Real-time and Coverage*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2007-05

I. Loeb. *Natural Deduction: Sharing by Presentation*. Faculty of Science, Mathematics and Computer Science, RU. 2007-06

M.W.A. Streppel. *Multifunctional Geometric Data Structures*. Faculty of Mathematics and Computer Science, TU/e. 2007-07

N. Trčka. *Silent Steps in Transition Systems and Markov Chains*. Faculty of Mathematics and Computer Science, TU/e. 2007-08

R. Brinkman. *Searching in encrypted data*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2007-09

A. van Weelden. *Putting types to good use*. Faculty of Science, Mathematics and Computer Science, RU. 2007-10

J.A.R. Noppen. *Imperfect Information in Software Development Processes*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2007-11

R. Boumen. *Integration and Test plans for Complex Manufacturing Systems*. Faculty of Mechanical Engineering, TU/e. 2007-12

A.J. Wijs. *What to do Next?: Analysing and Optimising System Behaviour in Time*. Faculty of Sciences, Division of Mathematics and Computer Science, VUA. 2007-13

C.F.J. Lange. *Assessing and Improving the Quality of Modeling: A Series of Empirical Studies about the UML*. Faculty of Mathematics and Computer Science, TU/e. 2007-14

T. van der Storm. *Component-based Configuration, Integration and Delivery*. Faculty of Natural Sciences, Mathematics, and Computer Science,UvA. 2007-15

B.S. Graaf. *Model-Driven Evolution of Software Architectures*. Faculty of Electrical Engineering, Mathematics, and Computer Science, TUD. 2007-16

A.H.J. Mathijssen. *Logical Calculi for Reasoning with Binding*. Faculty of Mathematics and Computer Science, TU/e. 2007-17

D. Jarnikov. *QoS framework for Video Streaming in Home Networks*. Faculty of Mathematics and Computer Science, TU/e. 2007-18

M. A. Abam. *New Data Structures and Algorithms for Mobile Data*. Faculty of Mathematics and Computer Science, TU/e. 2007-19

W. Pieters. *La Volonté Machinale: Understanding the Electronic Voting Controversy*. Faculty of Science, Mathematics and Computer Science, RU. 2008-01

A.L. de Groot. *Practical Automaton Proofs in PVS*. Faculty of Science, Mathematics and Computer Science, RU. 2008-02

M. Bruntink. *Renovation of Idiomatic Crosscutting Concerns in Embedded Systems*. Faculty of Electrical Engineering, Mathematics, and Computer Science, TUD. 2008-03

A.M. Marin. *An Integrated System to Manage Crosscutting Concerns in Source Code*. Faculty of Electrical Engineering, Mathematics, and Computer Science, TUD. 2008-04

N.C.W.M. Braspenning. *Model-based Integration and Testing of High-tech Multi-disciplinary Systems*. Faculty of Mechanical Engineering, TU/e. 2008-05

M. Bravenboer. *Exercises in Free Syntax: Syntax Definition, Parsing, and Assimilation of Language Conglomerates*. Faculty of Science, UU. 2008-06

M. Torabi Dashti. *Keeping Fairness Alive: Design and Formal Verification of Optimistic Fair Exchange Protocols*. Faculty of Sciences, Division of Mathematics and Computer Science, VUA. 2008-07

I.S.M. de Jong. *Integration and Test Strategies for Complex Manufacturing Machines*. Faculty of Mechanical Engineering, TU/e. 2008-08

I. Hasuo. *Tracing Anonymity with Coalgebras*. Faculty of Science, Mathematics and Computer Science, RU. 2008-09

L.G.W.A. Cleophas. *Tree Algorithms: Two Taxonomies and a Toolkit*. Faculty of Mathematics and Computer Science, TU/e. 2008-10

I.S. Zapreev. *Model Checking Markov Chains: Techniques and Tools*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2008-11

M. Farshi. *A Theoretical and Experimental Study of Geometric Networks*. Faculty of Mathematics and Computer Science, TU/e. 2008-12

G. Gulesir. *Evolvable Behavior Specifications Using Context-Sensitive Wildcards*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2008-13

F.D. Garcia. *Formal and Computational Cryptography: Protocols, Hashes and Commitments*. Faculty of Science, Mathematics and Computer Science, RU. 2008-14

P. E. A. Dürr. *Resource-based Verification for Robust Composition of Aspects*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2008-15

E.M. Bortnik. *Formal Methods in Support of SMC Design*. Faculty of Mechanical Engineering, TU/e. 2008-16

R.H. Mak. *Design and Performance Analysis of Data-Independent Stream*

Processing Systems. Faculty of Mathematics and Computer Science, TU/e. 2008-17

M. van der Horst. *Scalable Block Processing Algorithms*. Faculty of Mathematics and Computer Science, TU/e. 2008-18

C.M. Gray. *Algorithms for Fat Objects: Decompositions and Applications*. Faculty of Mathematics and Computer Science, TU/e. 2008-19

J.R. Calamé. *Testing Reactive Systems with Data - Enumerative Methods and Constraint Solving*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2008-20

E. Mumford. *Drawing Graphs for Cartographic Applications*. Faculty of Mathematics and Computer Science, TU/e. 2008-21

E.H. de Graaf. *Mining Semi-structured Data, Theoretical and Experimental Aspects of Pattern Evaluation*. Faculty of Mathematics and Natural Sciences, UL. 2008-22

R. Brijder. *Models of Natural Computation: Gene Assembly and Membrane Systems*. Faculty of Mathematics and Natural Sciences, UL. 2008-23

A. Koprowski. *Termination of Rewriting and Its Certification*. Faculty of Mathematics and Computer Science, TU/e. 2008-24

U. Khadim. *Process Algebras for Hybrid Systems: Comparison and Development*. Faculty of Mathematics and Computer Science, TU/e. 2008-25

J. Markovski. *Real and Stochastic Time in Process Algebras for Performance Evaluation*. Faculty of Mathematics and Computer Science, TU/e. 2008-26

H. Kastenberg. *Graph-Based Software Specification and Verification*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2008-27

I.R. Buhan. *Cryptographic Keys from Noisy Data Theory and Applications*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2008-28

R.S. Marin-Perianu. *Wireless Sensor Networks in Motion: Clustering Algorithms for Service Discovery and Provisioning*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2008-29

M.H.G. Verhoef. *Modeling and Validating Distributed Embedded Real-Time Control Systems*. Faculty of Science, Mathematics and Computer Science, RU. 2009-01

M. de Mol. *Reasoning about Functional Programs: Sparkle, a proof assistant for Clean*. Faculty of Science, Mathematics and Computer Science, RU. 2009-02

M. Lormans. *Managing Requirements Evolution*. Faculty of Electrical Engineering, Mathematics, and Computer Science, TUD. 2009-03

M.P.W.J. van Osch. *Automated Model-based Testing of Hybrid Systems*. Faculty of Mathematics and Computer Science, TU/e. 2009-04

H. Sozer. *Architecting Fault-Tolerant Software Systems*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2009-05

M.J. van Weerdenburg. *Efficient Rewriting Techniques*. Faculty of Mathematics and Computer Science, TU/e. 2009-06

H.H. Hansen. *Coalgebraic Modelling: Applications in Automata Theory and*

Modal Logic. Faculty of Sciences, Division of Mathematics and Computer Science, VUA. 2009-07

A. Mesbah. *Analysis and Testing of Ajax-based Single-page Web Applications*. Faculty of Electrical Engineering, Mathematics, and Computer Science, TUD. 2009-08

A.L. Rodriguez Yakushev. *Towards Getting Generic Programming Ready for Prime Time*. Faculty of Science, UU. 2009-9

K.R. Olmos Joffré. *Strategies for Context Sensitive Program Transformation*. Faculty of Science, UU. 2009-10

J.A.G.M. van den Berg. *Reasoning about Java programs in PVS using JML*. Faculty of Science, Mathematics and Computer Science, RU. 2009-11

M.G. Khatib. *MEMS-Based Storage Devices. Integration in Energy-Constrained Mobile Systems*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2009-12

S.G.M. Cornelissen. *Evaluating Dynamic Analysis Techniques for Program Comprehension*. Faculty of Electrical Engineering, Mathematics, and Computer Science, TUD. 2009-13

D. Bolzoni. *Revisiting Anomaly-based Network Intrusion Detection Systems*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2009-14

H.L. Jonker. *Security Matters: Privacy in Voting and Fairness in Digital Exchange*. Faculty of Mathematics and Computer Science, TU/e. 2009-15

M.R. Czenko. *TuLiP - Reshaping Trust Management*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2009-16

T. Chen. *Clocks, Dice and Processes*. Faculty of Sciences, Division of Mathematics and Computer Science, VUA. 2009-17

C. Kaliszyk. *Correctness and Availability: Building Computer Algebra on top of Proof Assistants and making Proof Assistants available over the Web*. Faculty of Science, Mathematics and Computer Science, RU. 2009-18

R.S.S. O'Connor. *Incompleteness & Completeness: Formalizing Logic and Analysis in Type Theory*. Faculty of Science, Mathematics and Computer Science, RU. 2009-19

B. Ploeger. *Improved Verification Methods for Concurrent Systems*. Faculty of Mathematics and Computer Science, TU/e. 2009-20

T. Han. *Diagnosis, Synthesis and Analysis of Probabilistic Models*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2009-21

R. Li. *Mixed-Integer Evolution Strategies for Parameter Optimization and Their Applications to Medical Image Analysis*. Faculty of Mathematics and Natural Sciences, UL. 2009-22

J.H.P. Kwisthout. *The Computational Complexity of Probabilistic Networks*. Faculty of Science, UU. 2009-23

T.K. Cocx. *Algorithmic Tools for Data-Oriented Law Enforcement*. Faculty of Mathematics and Natural Sciences, UL. 2009-24